

IMPLEMENTASI *LOW POWER* PADA *EMBEDDED SYSTEM* UNTUK MENDETEKSI KONDISI KEBAKARAN DALAM RUANGAN

SKRIPSI

KEMINATAN TEKNIK KOMPUTER

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Rinaldi Albert Soritua
NIM: 135150301111118



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018**

PENGESAHAN

IMPLEMENTASI *LOW POWER* PADA EMBEDDED SYSTEM UNTUK MENDETEKSI
KONDISI KEBAKARAN DALAM RUANGAN

SKRIPSI

KEMINATAN TEKNIK KOMPUTER

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :

Rinaldi Albert Soritua

NIM: 135150301111118

Skripsi ini telah diuji dan dinyatakan lulus pada
30 Juli 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Mochammad Hannats Hanafi Ichsan, S.ST., M.T

NIK: 201405 881229 1 001

Gembong Edhi Setyawan, S.T., M.T

NIK: 201208 761201 1 001

Mengetahui

Ketua Jurusan Teknik Informatika

Tri Astoto Kurniawan, S.T., M.T., Ph.D

NIP: 19710518 200312 1 001

PERNYATAAN ORISINALITAS

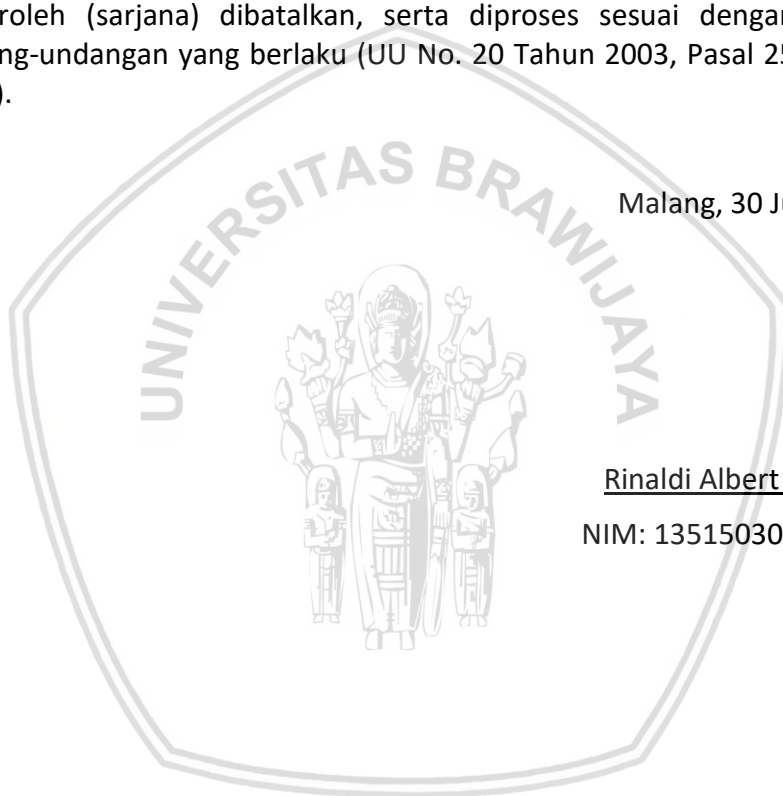
Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 30 Juli 2018

Rinaldi Albert Soritua

NIM: 135150300111011



KATA PENGANTAR

Puji syukur saya panjatkan kehadirat Tuhan Yang Maha Esa, karena Rahmat-Nya penulis dapat menyelesaikan skripsi ini. Skripsi yang berjudul “Implementasi *Low Power* Pada Embedded System Untuk Mendeteksi Kondisi Kebakaran Dalam Ruangan” ini disusun untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Komputer.

Penulis menyadari betul bahwa dalam penyusunan skripsi ini tidak terlepas dari bantuan berbagai pihak. Oleh karena itu penulis mengucapkan terima kasih kepada berbagai pihak yang telah berkenan untuk memberikan bantuan demi kelancaran penyusunan skripsi ini diantaranya:

1. Kedua orang tua penulis R.B.Siburian dan M.Rajaguk-guk yang tidak henti-henti memberikan dukungan, semangat dan do’a.
2. Saudara penulis : Sixka Novalita dan Jusuf Fernando Hasiholan
3. Bapak Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D. selaku Dekan Fakultas Ilmu Komputer Universitas Brawijaya Malang
4. Bapak Heru Nurwasito, Ir., M.Kom. selaku Wakil Dekan I Bidang Akademik Fakultas Ilmu Komputer Universitas Brawijaya Malang
5. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D. selaku Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya Malang
6. Bapak Mochammad Hannats Hanafi Ichsan, S.ST, M.T. selaku dosen pembimbing I yang telah memberikan pengarahan dalam pembuatan proposal dan alat dalam skripsi.
7. Bapak Gembong Edhi Setyawan, S.T, M.T. selaku dosen pembimbing II yang telah memberikan pengarahan dalam penulisan skripsi.
8. Segenap dosen Fakultas Ilmu Komputer Universitas Brawijaya atas segenap ilmu pengetahuan dan perhatian yang diberikan segenap staff dan pegawai Fakultas Ilmu Komputer Universitas Brawijaya atas segala bantuan yang bersifat administratif.
9. Teman-teman Kontrakan Ceria : Vatikan Aulia Makkah, Fauzi Rivani, Pramudya Mahardika, Rioadam, Ikhwan Zulfy, Agung Prasetio, Ahmad Faris, Alvin Leonardo, Arief Kurniawan, Fachrur, Fadlun Akbar, Garesi Putra, Aditya Geraldo, Nanda Epriliana, Nashir Umam Hasbi, Putri Meidyta, Rando, Ridzhal Hachim, Rint Zata Amani, Shelsa, Shinta, Sofi Hanifah, Ivana Agustina, dan Syahriel.
10. Ayu Samura, Eko Hilmy, Ihsannurrahim, Rezak Andri Purnomo, Lavanna Indanus, Oktaviany Setyowati, Putri Laras Rinjani, Irwanda Adhi F, Arleen Maranatha, Andrew Aditya.
11. Seluruh civitas akademika Informatika Universitas Brawijaya dan teman – teman Teknik Komputer Angkatan 2013 yang telah mendukung dan

memberi arahan selama peneliti menempuh studi di Teknik Komputer Universitas Brawijaya dan selama penyelesaian skripsi.

12. Seluruh pihak yang tidak dapat disebutkan satu persatu dalam membantu dan memberi arahan, peneliti sangat mengucapkan terima kasih sebanyak – banyak nya dalam bentuk dukungan dan do'a sehingga laporan skripsi ini terselesaikan.
13. Peneliti sangat mengharap kritik dan saran yang membangun, karena peneliti menyadari banyak kekurangan dalam menyusun laporan skripsi ini. Harapan dari penulis agar penelitian ini dapat bermanfaat bagi penelitian selanjutnya dan perkembangan teknologi yang sedang dikembangkan.

Malang, 30 Juli 2018

Penulis

rinaldialbs@gmail.com



ABSTRAK

Rinaldi Albert, Implementasi Low Power Pada Embedded System Untuk Mendeteksi Kondisi Kebakaran Dalam Ruangan

Pembimbing : Mochammad Hannats Hanafi Ichsan, S.ST, M.T dan Gembong Edhi Setyawan, S.T, M.T

Kebakaran merupakan suatu kejadian dimana salah satu faktor terjadinya kebakaran yaitu *human error*. Kebakaran dapat menimbulkan bermacam-macam kerugian mulai dari harta benda sampai korban jiwa. Berdasarkan permasalahan tersebut dibutuhkan sebuah sistem yang dapat mendeteksi terjadinya kebakaran berdasarkan kadar asap, suhu ruangan, dan keberadaan sumber api. Dalam penerapannya dibutuhkan kinerja *low power* untuk menghemat konsumsi daya ketika sistem tidak mendeteksi adanya kebakaran. *Output* dari sistem adalah tidak kebakaran, berasap, berapi asap sedikit, dan berapi asap banyak. *Output* ditampilkan pada LCD dan bunyi dari *Buzzer*. Mikrokontoller Arduino Mega 2560 ditanamkan logika *fuzzy sugeno* sebagai pemberi keputusan *output* berdasarkan hitungan fuzzy. Apabila sistem mendeteksi tidak ada kebakaran maka sistem akan beralih kedalam mode *sleep*, namun jika sistem mendeteksi adanya *output* berupa berasap, berapi asap sedikit, dan berapi asap banyak sistem akan menampilkan pada LCD dan bunyi dari *buzzer*. Dengan memanfaatkan kinerja *low power* pada mikrokontroller Arduino Mega 2560 diharapkan memberikan implementasi konsumsi daya yang lebih hemat. Fitur yang digunakan yakni *sleep mode power down* dimana sistem mematikan beberapa sub sistem yang ada. Pada pengujian sistem dapat dikatakan dapat berjalan sesuai dengan prinsip kerjanya. Pada pembacaan sensor MQ-2 sensor dapat mendeteksi kadar asap yang terdeteksi pada sensor. Pada pembacaan sensor DHT11 dapat membaca suhu ruangan dengan tingkat error 3%. Pada pembacaan sensor *flame* dapat membaca keberadaan api melalui jarak yang bervariasi. Sistem dapat menentukan berbagai kondisi dengan keakuratan fuzzy sebesar 100%. Rata-rata waktu eksekusi sistem adalah $\pm 33284,2$ ms. Pada konsumsi daya perangkat mampu menekan arus sebesar 40mA dari 245mA dengan penerapan *sleep mode power down*.

Kata Kunci : Kebakaran, *Low Power*, Fuzzy Sugeno.

ABSTRACT

Rinaldi Albert, Implementation of Low Power on the Embedded System to Detect Indoor Fire Conditions

Pembimbing : Mochammad Hannats Hanafi Ichsan, S.ST, M.T dan Gembong Edhi Setyawan, S.T, M.T

Fire is an event which one of the factor is caused by human error. Fire could cause various losses from treasures to casualties. Based from that problem needed a system that able to detect fire based on smoke level, room temperature, dan existence fire source. In implementation needed low power to save power consumption when system not detect a fire. The output of the system is no fire, fire and smoky little, fire smoky and lot that shown on LCD and sound from buzzer. Arduino Mega 2560 implanted fuzzy logic as output decision based on fuzzy calculation. If system not detect a fire then system will switch to sleep mode, yet if system detect an output which is smoky, fire and smoky little, fire smoky lot it will shown on LCD and sound from buzzer. Utilize low power performance at microcontroller Arduino Mega 2560 expected give implementation on power consumption more efficient. The used feature is sleep mode power down to turn off some sub-system. Testing resulted system is running accordance with the principles of it works. On readings of MQ-2 sensor could detect smoke level on sensor. On readings of DHT11 could read temperature of room with error rate 3%. On readings of flame sensor could detect fire existence through various distance. System could determine various condition with 100% fuzzy accuracy. The average system execution is $\pm 33284,2$ ms. On the power consumption of the device is able pressed for 40mA from 245mA to 205mA with implementation sleep mode power down.

Keywords : Fire, Low Power, Fuzzy Sugeno

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR	xii
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah	2
1.3 Tujuan	3
1.4 Manfaat.....	3
1.5 Batasan masalah	3
1.6 Sistematika pembahasan.....	4
BAB 2 LANDASAN KEPUSTAKAAN	6
2.1 Kajian Pustaka	6
2.2 Dasar Teori	6
2.2.1 Bahaya Kebakaran.....	7
2.2.2 Arduino IDE	7
2.2.3 Sleep Mode	7
2.2.4 Fuzzy Logic.....	8
2.2.5 Persentase dan rata-rata kesalahan	9
2.2.6 Perhitungan pemakaian daya	9
2.2.7 Persentase penurunan daya	9
2.2.8 ADC.....	10
BAB 3 METODOLOGI	11
3.1 Studi literatur	11
3.2 Rekayasa Kebutuhan.....	11
3.3 Perancangan	12

3.4 Implementasi	12
3.4.1 Impementasi Perangkat Keras	12
3.4.2 Implementasi Perangkat Lunak.....	12
3.5 Pengujian dan Analisis	13
3.6 Kesimpulan.....	13
BAB 4 REKAYASA KEBUTUHAN	14
4.1 Gambaran Umum Sistem.....	14
4.2 Analisis Kebutuhan Sistem.....	14
4.2.1 Kebutuhan Fungsional.....	14
4.2.2 Kebutuhan Non-Fungsional	15
4.3 Batasan Desain Sistem	19
BAB 5 PERANCANGAN DAN IMPLEMENTASI	20
5.1 Perancangan Sistem.....	20
5.1.1 Perancangan Perangkat Keras	20
5.1.2 Perancangan Perangkat Lunak.....	21
5.2 Implementasi Sistem	36
5.2.1 Implementasi Perangkat Keras	36
5.2.2 Implementasi Perangkat Lunak.....	36
BAB 6 PENGUJIAN DAN ANALISIS.....	45
6.1 Pengujian Akuisisi Data Sensor MQ-2.....	46
6.1.1 Tujuan.....	46
6.1.2 Prosedur	46
6.1.3 Hasil dan Analisis.....	46
6.2 Pengujian Akuisisi Data Sensor DHT11	48
6.2.1 Tujuan.....	48
6.2.2 Prosedur	48
6.2.3 Hasil dan Analisis.....	48
6.3 Pengujian Akuisisi Data Sensor <i>Flame</i>	49
6.3.1 Tujuan.....	49
6.3.2 Prosedur	49
6.3.3 Hasil dan Analisis.....	50
6.4 Pengujian Proses Fuzzy	51

6.4.1 Tujuan.....	51
6.4.2 Prosedur	51
6.4.3 Hasil dan Analisis.....	51
6.5 Pengujian Low Power.....	52
6.5.1 Tujuan Pengujian.....	52
6.5.2 Prosedur	53
6.5.3 Hasil dan Analisis.....	53
6.6 Pengujian Waktu Eksekusi Sistem	54
6.6.1 Tujuan.....	54
6.6.2 Prosedur	54
6.6.3 Hasil Pengujian	55
BAB 7 PENUTUP	56
7.1 Kesimpulan.....	56
7.2 Saran	56
DAFTAR PUSTAKA.....	57



DAFTAR TABEL

Tabel 4.1 Spesifikasi Sensor DHT11	16
Tabel 4.2 Spesifikasi Sensor Flame.....	18
Tabel 4.3 Deskripsi Pin Buzzer.....	18
Tabel 5.1 Koneksi Pin Perancangan Perangkat Keras	21
Tabel 5.2 Pembuatan Rule Fuzzy	32
Tabel 5.3 Kode Program Fungsi Sleepnow()	37
Tabel 5.4 Kode Program Kalibrasi Sensor MQ-2	38
Tabel 5.5 Kode Program Fungsi MQCalibration	38
Tabel 5.6 Kode Program Fungsi MQRead	39
Tabel 5.7 Kode Program Fungsi MQResistanceCalculation	39
Tabel 5.8 Kode Program Fungsi MQGetGasPercentage	40
Tabel 5.9 Kode Program Kalibrasi ADC ke PPM	40
Tabel 5.10 Kode Program Fuzzifikasi Suhu	40
Tabel 5.11 Kode Program Fuzzifikasi Asap	41
Tabel 5.12 Kode Program Fuzzifikasi Api	42
Tabel 5.13 Kode Program Pembuatan Rule	42
Tabel 5.14 Kode Program Defuzzifikasi.....	43
Tabel 5.15 Kode Program Output LCD dan Buzzer	44
Tabel 6.1 Hasil Pengujian Sensor MQ-2 melalui Arduino IDE	46
Tabel 6.2 Analisis sistem error pada sensor DHT11.....	48
Tabel 6.3 Nilai Pembacaan Analog pada Sensor <i>Flame</i>	50
Tabel 6.4 Hasil Pengujian Fuzzy	52
Tabel 6.6 Hasil Pengujian Konsumsi Arus Listrik	54
Tabel 6.7 Hasil Pengujian Waktu.....	55

DAFTAR GAMBAR

Gambar 2.1 Mode pada <i>Sleep Mode</i>	8
Gambar 3.1 Diagram Alir Metodologi Penelitian.....	11
Gambar 3.2 Diagram Blok Sistem	12
Gambar 4.1 Arduino Mega 2560.....	15
Gambar 4.2 Spesifikasi Arduino Mega 2560 dan Arduino Uno	16
Gambar 4.3 Sensor DHT11	16
Gambar 4.4 Sensor MQ-2.....	17
Gambar 4.5 Spesifikasi Sensor MQ-2	17
Gambar 4.6 Sensor <i>Flame</i>	17
Gambar 4.7 LCD 2x16.....	18
Gambar 4.8 Buzzer	19
Gambar 5.1 Skema Perancangan Perangkat Keras.....	20
Gambar 5.2 Flowchart perancangan perangkat lunak sistem	22
Gambar 5.3 Flowchart perancangan sensor <i>flame</i> sebagai pin interrupt.....	24
Gambar 5.4 Flowchart perancangan perangkat lunak sensor asap MQ-2	24
Gambar 5.5 Flowchart Perangkat Lunak Sensor Suhu	25
Gambar 5.6 Flowchart perancangan perangkat lunak <i>sleep mode power down</i> .	26
Gambar 5.7 <i>Flowchart</i> Perancangan <i>Fuzzy</i>	27
Gambar 5.8 <i>Flowchart</i> Proses Fuzzifikasi.....	28
Gambar 5.9 <i>Membership Function</i> Suhu	29
Gambar 5.10 <i>Membership Function</i> Asap.....	30
Gambar 5.11 <i>Membership Function</i> Api	31
Gambar 5.12 Flowchart Proses Inferensi	34
Gambar 5.13 Flowchart Proses Defuzzifikasi	35
Gambar 5.14 Tampilan prototipe dari depan	36
Gambar 5.15 Tampilan Prototipe dari dalam	36
Gambar 5.16 Deklarasi Pin Interrupt	37
Gambar 6.1 Diagram Blok Pengujian dan Analisis	45
Gambar 6.2 Grafik perbandingan Sensor DHT11 dan Thermometer Ruang	49
Gambar 6.3 Grafik Pengujian Pembacaan Sensor <i>Flame</i>	50

BAB 1 PENDAHULUAN

1.1 Latar belakang

Kebakaran merupakan suatu peristiwa yang disebabkan oleh api dan pembakaran tidak terkawal yang membahayakan nyawa manusia, bangunan dan lingkungan sekitar. Kebakaran berdasarkan penyebabnya dapat dibagi menjadi 3 bagian, yaitu kelas A yang disebabkan oleh benda-benda padat seperti kertas, kayu, karet, plastik, dsb. Kemudian pada kelas B yaitu kebakaran yang disebabkan oleh cairan yang mudah terbakar seperti bensin, solar, minyak tanah, dan spritus. Sedangkan kelas C yang disebabkan oleh listrik (Perda, 1992).

Kebakaran sering terjadi pada area padat penduduk dikarenakan adanya kelalaian oleh pemilik rumah yang lupa mematikan alat elektronik ataupun ketika pemilik rumah sedang tidak ada di rumah. Pada umumnya kebakaran yang sering terjadi pada area perumahan adalah hubungan arus pendek listrik. Di kawasan kali baru, Clincing, Jakarta Utara kebakaran terjadi pada area padat penduduk. Kebakaran tersebut memakan satu korban jiwa dan menghancurkan 16 petak rumah kontrakan dan 13 kios (Yusuf, 2017). Kejadian ini terjadi pada pukul 02.50 WIB dimana pemilik rumah sedang beristirahat dan tertidur lelap. Hal ini menyebabkan pemilik rumah terlambat untuk menyadari titik api yang berpotensi menyebabkan kebakaran. Hal ini juga yang menyebabkan terjadinya korban jiwa ketika tidak ada peringatan kebakaran yang menyebabkan pemilik rumah terlambat untuk mengantisipasinya. Bagian rumah yang berpotensi menyebabkan kebakaran salah satunya adalah dapur. Dapur merupakan tempat memasak yang memungkinkan terjadinya percikan api ketika sedang memasak dan banyaknya bahan-bahan yang mudah terbakar seperti minyak goreng, tabung gas LPG, dan lain-lain. Pemicu kebakaran yang sering terjadi di dapur adalah berasal dari gas bocor atau kompor yang lupa dimatikan oleh pemilik rumah meninggalkan rumah.

Untuk meminimalisir potensi terjadinya kebakaran yang dapat mengakibatkan jatuhnya korban jiwa dan juga materi diperlukan sebuah sistem yang dapat mendeteksi kebakaran sebagai peringatan dini jika terdapat potensi akan terjadinya kebakaran. Didalam kebakaran terdapat 3 elemen yaitu bahan bakar, suhu/panas, dan oksigen yang kemudian akan membentuk api. 3 elemen tersebut disebut dengan segitiga api (*fire triangle*) (Pacific, 2017). Pada teori *fire triangle* apabila salah satu unsur dari 3 elemen api tidak terpenuhi maka api tidak akan terbentuk dan peristiwa kebakaran tidak akan terjadi. Untuk mengatasi masalah tersebut dilakukan penelitian lebih lanjut dengan menggunakan sensor *flame* berfungsi untuk mendeteksi api, sensor DHT11 berfungsi untuk mendeteksi suhu, dan sensor MQ-2 berfungsi untuk mendeteksi asap dan metode *low power* agar sistem berada pada kondisi *idle* apabila tidak ada asap, suhu, dan api yang terdeteksi. Low power dibutuhkan dalam sistem ini dikarenakan pola konsumsi energy di Indonesia telah masuk kedalam kategori boros. Salah satu penyebab borosnya penggunaan energy di Indonesia adalah penggunaan alat elektronik dan sistem elektronik yang tidak efisien dan tidak sesuai dengan kebutuhan yang

dipakai. Dari laporan ACEEE (*American Council for An Energy-Efficient Economy*) Indonesia berada pada peringkat ke 18 dari 23 negara yang terdaftar sebagai pengguna dan konsumsi listrik tertinggi di dunia (ACEEE, 2016). Maka dari itu *Low Power* berfungsi untuk menghemat pemakaian daya listrik yang sia-sia. Pada alat-alat pendeteksi kebakaran yang sudah ada dipasaran saat ini, menggunakan prinsip kerja dengan hanya menggunakan 2 pendeteksi kebakaran yaitu suhu dan asap. Ketika alat mendeteksi adanya kenaikan suhu dan asap yang terdeteksi maka alat akan memberikan peringatan kepada pemilik rumah. Pada penelitian sebelumnya oleh (Nurromianto, 2015) hanya memakai sensor asap dan suhu. Namun pada penelitian (Purnomo, 2017) terdapat penambahan sensor berupa sensor flame yang dapat mendeteksi adanya api.

Maka dibutuhkan adanya Implementasi *Low Power* pada alat pendeteksi kebakaran, untuk mengurangi pemakaian daya listrik yang berlebihan. Implementasi *low power mode* disini memungkinkan menjalankan sistem dengan efektifitas tegangan yang lebih rendah, clock speed yang lebih rendah, dan konsumsi daya yang lebih hemat guna untuk mengefisienkan daya pada sistem.

Pada dasarnya fitur *low power* terbagi atas beberapa fitur, yakni *Idle*, *Power-down*, *Power-save*, *Standby*, *Extended Standby*. Dimana dalam penggunaannya fitur *Power-down* memiliki penghematan daya lebih besar dikarenakan memutuskan tegangan pada ADC (*Analog Digital Converter*) dan juga mematikan BoD (*Brown-out Detection*) sehingga meringankan beban konsumsi arus pada sistem.

Pada penelitian ini penulis melakukan pengembangan terhadap penelitian sebelumnya yang dilakukan oleh (Purnomo, 2017) dengan menambahkan *low power* terhadap sistem dan menggunakan fitur *Power-Down* untuk melakukan penghematan daya dalam sistem.

Oleh karena itu, penulis membuat skripsi tentang Implementasi *Low Power* Pada Embedded System Untuk Mendeteksi Kondisi Kebakaran Dalam Ruangan dengan menggunakan 3 sensor yaitu sensor DHT11, sensor MQ-2, dan sensor *flame*. Data dari sensor-sensor akan diolah oleh *mikroprosesor* namun pada saat sensor tidak mendeteksi ada nya kenaikan suhu, asap dan api yang terdeteksi maka sistem akan berada pada *sleep mode power down* untuk penghematan daya. Output yang digunakan pada penelitian yang akan dilakukan adalah dengan menggunakan LCD 16x4 dan Buzzer. LCD digunakan untuk memberikan teks kondisi ruangan yang dibaca oleh sensor. Sedangkan *buzzer* digunakan untuk memberikan alarm sebagai peringatan tanda bahaya.

1.2 Rumusan masalah

Berdasarkan latar belakang di atas dapat dirumuskan masalah yang akan diangkat pada penelitian, yaitu:

1. Bagaimana merancang sistem pendeteksi dalam ruangan berdasarkan parameter suhu, asap, dan api menggunakan Arduino Mega 2560?

2. Bagaimana merancang dan mengimplementasikan mekanisme *low power* kepada sistem pendeteksi kebakaran dalam ruangan menggunakan Arduino Mega 2560?
3. Bagaimana performa mekanisme *low power mode* yang dijalankan terhadap sistem pendeteksi kebakaran dalam ruangan menggunakan Arduino Mega 2560?

1.3 Tujuan

Dari rumusan masalah di atas didapatkan tujuan dari penelitian, yaitu:

1. Merancang dan mengimplementasikan sistem deteksi kebakaran dengan menggunakan *mikrokontroller Arduino Mega*, sensor suhu DHT11, sensor asap MQ-2, dan sensor *flame*.
2. Merancang dan mengimplementasikan mekanisme *low power mode* kedalam sistem deteksi kebakaran.
3. Dapat mengukur performa mekanisme *low power mode* terhadap sistem.

1.4 Manfaat

Adapun manfaat yang diperoleh dari hasil penelitian ini yakni sebagai berikut :

1. Sistem ini diharapkan dapat memonitoring potensi kebakaran dalam ruangan dengan baik.
2. Implementasi *Low Power Mode* diharapkan dapat memberikan solusi dalam pemakaian daya yang tinggi menjadi lebih rendah.
3. Memberikan alternatif dalam penggunaan fitur-fitur yang terdapat dalam *Low Power* agar dapat diimplementasikan dalam pengkondisian *low energy*.
4. Mengetahui beberapa sumber pengetahuan mengenai penghematan energi pada piranti baik dari perangkat lunak maupun perangkat keras.

1.5 Batasan masalah

Agar dalam penelitian ini dapat lebih fokus dalam pengerjaannya maka diberikan batasan masalah sebagai berikut:

1. Sistem masih berupa prototipe.
2. Implementasi metode *low power* menggunakan *Arduino Mega*.
3. Sistem menggunakan metode *fuzzy sugeno* dalam pengolahan data.
4. Sensor yang digunakan sebanyak 4 sensor, yakni sensor Suhu DHT11, sensor Asap MQ-2, dan 2 buah sensor *flame*.
5. Metode *low power* yang digunakan menggunakan *sleep mode* dan *wake*.
6. Mode *sleep* yang digunakan yakni *sleep mode power down*.
7. Mode *wake* yang digunakan yakni *external interrupt*.
8. Keluaran ditampilkan pada LCD dan *buzzer* sebagai peringatan.

1.6 Sistematika pembahasan

Sistematika penulisan penelitian ditunjukkan untuk memberikan gambaran dan uraian dari penyusunan skripsi secara garis besar yang meliputi beberapa bab, sebagai berikut :

BAB I : Pendahuluan

Bab ini menjelaskan dasar dari penelitian yang terdiri dari latar belakang terjadinya kebakaran pada ruangan dan pemakaian daya yang berlebihan, mengidentifikasi masalah, rumusan masalah, tujuan, manfaat dilakukannya penelitian, batasan masalah dan sistematika pembahasan dari penelitian yang akan dilakukan yaitu “Implementasi *Low Power* Pada Embedded System Untuk Mendeteksi Kondisi Kebakaran Dalam Ruangan”.

BAB II : Landasan Kepustakaan

Bab ini menjelaskan landasan kepustakaan dari penelitian yang dibuat yaitu “Implementasi *Low Power* Pada Embedded System Untuk Mendeteksi Kondisi Kebakaran Dalam Ruangan” yang berisi kajian-kajian bahan pustaka penelitian sebelumnya dari berbagai refrensi seperti jurnal, paper, dan sumber lainnya terkait *low power* beserta dasar tentang fuzzy.

BAB III : Metodologi

Bab ini menjelaskan tentang metode dan langkah alur kerja penelitian yang terdiri dari studi literatur, rekayasa kebutuhan yang terdiri dari kebutuhan fungsional dan kebutuhan non-fungsional, kebutuhan perangkat keras dan perangkat lunak, perancangan sistem yang dijelaskan pada blok diagram dan implementasi sistem.

BAB IV : Rekayasa Kebutuhan

Bab ini menjelaskan tentang kebutuhan-kebutuhan yang diperlukan untuk merancang “Implementasi *Low Power* Pada Embedded System Untuk Mendeteksi Kondisi Kebakaran Dalam Ruangan” seperti kebutuhan sistem, kebutuhan fungsional, kebutuhan perangkat keras, dan kebutuhan perangkat lunak agar sistem dapat berjalan dengan baik.

BAB V : Perancangan dan Implementasi

Bab ini menjelaskan tentang tahapan-tahapan dalam perancangan dan implementasi sistem untuk “Implementasi *Low Power* Pada Embedded System Untuk Mendeteksi

Kondisi Kebakaran Dalam Ruangan” dan membahas tentang *low power* ke dalam sistem yang digunakan.

BAB VI : Pengujian dan Analisis

Bab ini menjelaskan bagaimana pengujian dilakukan, skenario pengujian pada sub-sistem maupun sistem itu sendiri. Hasil dari pengujian dianalisis dan dijabarkan secara terurut dan menggunakan teks dan ilustrasi diagram.

BAB VII : Penutup

Memuat kesimpulan yang diperoleh dari hasil pengujian sistem, serta saran-saran untuk pengembangan lebih lanjut.



BAB 2 LANDASAN KEPUSTAKAAN

Bab ini berisi tinjauan pustaka yang meliputi tinjauan pustaka dari dasar teori yang diperlukan dalam penelitian. Tinjauan pustaka membahas terkait penelitian yang telah dilakukan sebelumnya. Dasar teori membahas teori yang diperlukan dalam penelitian.

2.1 Kajian Pustaka

Kajian pustaka berisi tentang referensi yang berhubungan dengan penelitian yang telah dilakukan oleh peneliti sebelumnya. Kajian pustaka yang dijadikan penulis sebagai sumber adalah penelitian yang telah dilakukan oleh (Purnomo, 2017). Pada penelitian tersebut menggunakan *Fuzzy Logic* untuk menentukan output dari 3 sensor yaitu sensor suhu DHT11, sensor asap MQ-2, dan sensor *flame* dimana output dari sistem terbagi atas 4 jenis kondisi, yaitu tidak ada kebakaran, berasap, berasap sedikit, berasap banyak.

Pada penelitian selanjutnya yaitu yang telah dilakukan oleh (Ihsannurrahim, 2017). Sistem berupa monitoring detak jantung manusia melalui *wearable device* yang diletakkan pada daerah pergelangan tangan. Sistem akan menampilkan detak jantung dalam bentuk *beat per minute* pada sebuah layar yang terdapat pada perangkat dan hasil pembacaan sistem dapat dikirimkan melalui Bluetooth ke *smartphone* android. Terdapat beberapa state dalam sistem ini dimana sistem akan berada pada dalam keadaan *sleep*, keadaan *standby* dan sebagai agar penggunaan daya dapat berkurang. Untuk berpindah dari keadaan *sleep* sistem menggunakan pembacaan dari sensor *gyroscope* sehingga sistem dapat berpindah state menjadi *wake*. Pada penelitian tersebut sistem dapat menurunkan konsumsi arus mencapai 75,06% dari nilai 31,68mA menjadi 7,9 mA.

Pada penelitian selanjutnya yang telah dilakukan oleh (Nallani & Hency, 2015). Sistem berupa monitoring kelembapan tanah, dimana sistem akan bekerja ketika kelembapan tanah dan suhu didalam tanah melebihi ambang dengan mengaktifkan keran air. Kemudian sistem akan mengirimkan pemberitahuan melalui sms dan GPS kepada pengguna.

Berdasarkan penelitian-penelitian yang ada diatas, penulis bermaksud untuk mengembangkan penelitian pertama dengan membuat sistem *embedded* yang dapat mendeteksi adanya kebakaran pada rumah dengan menggunakan metode yang ada pada penelitian kedua yaitu *low power*.

2.2 Dasar Teori

Dalam Sub bab ini akan dijelaskan referensi dasar teori sebagai pengetahuan dalam pembuatan penelitian meliputi macam-macam alat pengukuran, mikrokontroler.

2.2.1 Bahaya Kebakaran

Ada 2 jenis bahaya yang ditimbulkan sebagai akibat dari terjadinya suatu kebakaran yaitu keselamatan jiwa manusia dan kerugian material. Beberapa aspek penyelamatan lebih diarahkan dan diprioritaskan pada penyelamatan jiwa manusia terlebih dahulu, untuk kemudian meminimalkan kerugian pada tahap berikutnya. Konsep penanggulangan kebakaran (*fire safety*) yang utama adalah penyelamatan jiwa manusia. Bahaya keselamatan jiwa manusia pada peristiwa kebakaran dapat diklasifikasikan sebagai berikut (Rahman, 2004).

1. bahaya langsung
 - a. tersengat temperatur yang tinggi
 - b. keracunan asap
2. bahaya tidak langsung
 - a. terluka
 - b. terjatuh
 - c. terserang sakit
 - d. mengalami shock/serangan psikologis

2.2.2 Arduino IDE

Arduino IDE adalah lingkungan pengembangan arduino yang berisi teks editor untuk menuliskan kode, pesan area, teks console, toolbar dengan tombol fungsi umum, dan serangkaian menu. Terhubung ke perangkat keras yang terhubung ke perangkat arduino board dan berkomunikasi dengan board tersebut. Perangkat lunak yang dituliskan menggunakan arduino disebut sketch, sketch ini ditulis dalam editor teks. Teks editor memiliki fitur copy/paste dan mencari/mengganti teks. Area pesan memberikan umpan balik saat menyimpan dan mengeksport dan juga menampilkan kesalahan yang terdapat dalam kode program

Sekalipun Arduino bisa digunakan di berbagai macam sistem operasi, karena dalam penulisan laporan tugas akhir ini penulis menggunakan sistem operasi Windows, maka hanya akan dibahas sedikit tentang instalasi Arduino di Windows. Software Arduino bisa didapat di situs resmi Arduino secara gratis.

2.2.3 Sleep Mode

Arduino Memiliki beberapa mode dalam menjalankan sleep mode dan juga memiliki beberapa fitur untuk menjalankan mode wake. Berikut beberapa fitur untuk menjalankan mode wake yaitu dengan menggunakan internal interrupt, external interrupt, UART, Watchdog Timer. Internal interrupt akan terjadi didalam program itu sendiri sedangkan untuk external interrupt akan terjadi apabila adanya masukan melalui lingkungan sekitar. Sedangkan pada Watchdog Timer interrupt akan terjadi secara periodik atau dalam jangka waktu yang sudah ditentukan. Sedangkan pada UART interrupt akan terjadi apabila data diterima pada serial interface. Berikut beberapa mode pada sleep mode pada Gambar 2.1.

	Active Clock Domains					Oscillators		Wake-up Sources						
	clk _{CPU}	clk _{FLASH}	clk _{IO}	clk _{ADC}	clk _{ASY}	Main Clock Source Enabled	Timer Oscillator Enabled	INT1, INT0 and Pin Change	TWI Address Match	Timer2	SPM/EEPROM Ready	ADC	WDT	Other/O
Sleep Mode														
Idle			X	X	X	X	X ⁽²⁾	X	X	X	X	X	X	X
ADC Noise Reduction				X	X	X	X ⁽²⁾	X ⁽³⁾	X	X ⁽²⁾	X	X	X	
Power-down								X ⁽³⁾	X				X	
Power-save					X		X ⁽²⁾	X ⁽³⁾	X	X			X	
Standby ⁽¹⁾						X		X ⁽³⁾	X				X	

Gambar 2.1 Mode pada Sleep Mode

Sumber : (Homautomaion, 2014)

2.2.4 Fuzzy Logic

Logika fuzzy adalah suatu cara yang tepat untuk memetakan suatu ruang input ke dalam ruang output. Pada sistem yang rumit, pemanfaatan logika fuzzy (fuzzy logic) adalah salah satu solusi untuk menyelesaikan masalah. Sistem secara tradisional dirancang untuk sebagai kontrol keluaran tunggal yang berasal dari beberapa masukan yang tidak saling berhubungan. Karena ketidaktergantungan ini, penambahan masukan yang baru akan menyusahakan proses kontrol dan hal itu membutuhkan proses kalkulasi kembali dari semua fungsi. Kebalikannya, penambahan masukan baru pada sistem fuzzy adalah sistem yang bekerja berdasarkan prinsip-prinsip logika fuzzy, sehingga pada sistem ini hanya membutuhkan penambahan fungsi keanggotaan yang baru dan aturan-aturan yang berhubungan dengannya (Kusumadewi S. , 2003).

Pada umumnya, sistem fuzzy sangat cocok dimanfaatkan untuk penalaran pendekatan terutama untuk sistem yang mengatasi masalah-masalah yang sulit dijelaskan dengan menggunakan model matematis. contohnya, nilai masukan dan parameter sebuah sistem mempunyai sifat kurang akurat atau kurang jelas, hal ini menyebabkan sulit mendefinisikan model matematikanya.

Logika fuzzy memungkinkan nilai keanggotaan antara 0 dan 1, tingkat keabuan antara hitam dan putih, dan dalam bentuk linguistik, konsep tidak pasti seperti "sedikit", "lumayan", dan "sangat". Hal ini berhubungan dengan set fuzzy dan teori kemungkinan, yang diperkenalkan oleh Dr. Lotfi Zadeh dari Universitas California, Berkeley pada 1965 (Kusumadewi & Hartati, 2006)

2.2.4.1 Metode Fuzzy Inference System (FIS) Sugeno

Sugeno FIS pada umumnya hampir sama dengan metode mamdani dari beberapa metode. Dari segi proses inferensi, fuzzifikasi input dan pengaplikasian operator fuzzy kedua metode tersebut memiliki kesamaan. Namun perbedaan

utamanya adalah Sugeno memiliki output berupa linear atau konstan. Metode Sugeno memiliki 2 model yaitu model orde-0 dan model orde-1.

Keluaran aturan demikian bukan dalam bentuk fungsi keanggotaan, tapi suatu bilangan yang berubah secara linier terhadap variabel-variabel input, yaitu mengikuti suatu persamaan bidang $z = av + bw + c$. Jika $b = 0$, sistem inferensi dikatakan berorder satu dimana keluarannya mengikuti persamaan garis, yaitu $z = av + c$. Jika $a = b = 0$, sistem inferensi dikatakan berorder nol, karena keluarannya berupa sebuah bilangan konstan, yaitu $z = c$.

2.2.5 Persentase dan rata-rata kesalahan

Persentase kesalahan digunakan untuk membandingkan nilai pembacaan dengan nilai pasti. Perhitungan kesalahan dapat membantu untuk melihat seberapa dekat estimasi terhadap nilai riil. Rumus persentase kesalahan berguna untuk menentukan ketepatan dari perhitungan. Rumus yang digunakan sebagai berikut:

$$\text{kesalahan}(\%) = \frac{HT - HS}{HT} \times 100\% \quad (2.1)$$

Keterangan :

HT : Hasil dari pembacaan termometer ruangan

HS : Hasil dari pembacaan sensor DHT11

Dalam melakukan perhitungan kesalahan dibutuhkan beberapa percobaan dalam menentukan perbedaan antara nilai riil dengan nilai pembacaan. Maka dibutuhkan perhitungan rata-rata terhadap seluruh percobaan. Rumus yang digunakan sebagai berikut:

$$\text{Rata - rata} = \frac{\text{Jumlah Hasil}}{\text{Banyak Percobaan}} \quad (2.2)$$

2.2.6 Perhitungan pemakaian daya

Perhitungan pemakaian daya diperlukan untuk mengetahui perbedaan pemakaian daya pada saat sistem dalam keadaan *wake* dengan sistem dalam keadaan *sleep*. Dalam melakukan perhitungan pemakaian daya sistem selama beberapa detik dalam keadaan *wake* dan *sleep* dibutuhkan rumus perhitungan sebagai berikut:

$$\text{Pemakaian} = V \times I \times \frac{n}{3600} \quad (2.3)$$

Keterangan :

V : Volt yang digunakan

I : Arus yang terbaca pada multimeter

n : jumlah detik yang diperlukan

2.2.7 Persentase penurunan daya

Perhitungan persentase penurunan daya dibutuhkan dimana perhitungan penurunan daya untuk menentukan seberapa besar penurunan pemakaian daya ketika sistem dalam keadaan *wake* dan dalam keadaan *sleep*. Untuk menghitung persentase penurunan pemakaian berikut rumus yang digunakan:

$$\text{Penurunan (\%)} = \frac{AW-AS}{AW} \times 100\% \quad (2.4)$$

Keterangan :

AW : Arus Kondisi Wake

AS : Arus Kondisi Sleep

2.2.8 ADC

Sebuah ADC (Analog to Digital Converter) adalah sebuah rangkaian elektronik yang berfungsi mengubah sinyal analog (kontinyu) menjadi keluaran sinyal digital (diskrit). Pada dasarnya sebuah ADC adalah sebuah perangkat atau piranti elektronik yang berfungsi mengubah sebuah tegangan menjadi sebuah bilangan digital biner.

Resolusi dari sebuah converter menunjukkan banyaknya nilai diskrit yang dihasilkan pada tegangan tertentu. Resolusi pada umumnya dinyatakan dalam satuan bit (binary digit). Contohnya, sebuah ADC sebuah masukan analog menjadi salah satu dari 256 nilai diskrit mempunyai resolusi 8 bit karena $2^8 = 256$. Pada penelitian ini penulis menggunakan Arduino Mega 2560 yang memiliki resolusi bit sebesar 10 bit dimana $2^{10} = 1024$

Resolusi dapat dinyatakan secara elektrik dalam satuan volt. Dengan menggunakan rumus

$$ADC = \frac{v_{in}}{v_{ref}} \times \text{Bit Data} \quad (2.5)$$

Vin : Tegangan Input

Vref : Tegangan Refrensi

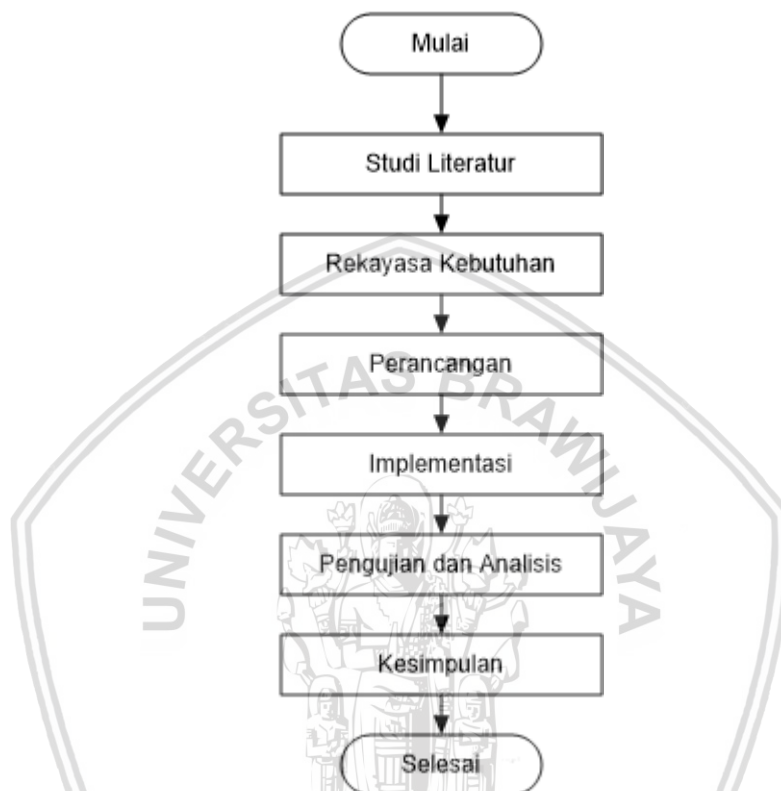
Bit Data : Bit Data Maksimal = $2^{(n-1)}$

Berikut adalah rumus untuk menghitung besarnya nilai tegangan ADC :

$$\text{Pembacaan Analog} \times \frac{\text{Jangkauan Tegangan}}{\text{Bit Data}} \quad (2.5)$$

BAB 3 METODOLOGI

Pada bab ini membahas tentang metode yang diterapkan pada skripsi ini yang meliputi : Studi literatur, Rekayasa Kebutuhan, gambaran umum perancangan, tahapan implementasi sistem, dan penarikan kesimpulan. Alur metodologi penelitian dalam skripsi ini digambarkan pada Gambar 3.1.



Gambar 3.1 Diagram Alir Metodologi Penelitian

3.1 Studi literatur

Studi literatur dilakukan bertujuan untuk mempelajari serta memahami penjelasan dasar teori yang digunakan untuk menunjang agar tidak mengalami kendala pada tahapan perancangan. Pada tahap studi literatur ini mempelajari teori-teori yang digunakan dalam pengerjaan skripsi. Di antara teori pendukung tersebut diperoleh dari buku, jurnal, e-book, dokumentasi, dan penelitian yang berkaitan dengan skripsi. Referensi utama yang diperlukan dalam penulisan ini adalah *forum Arduino dan jurnal*.

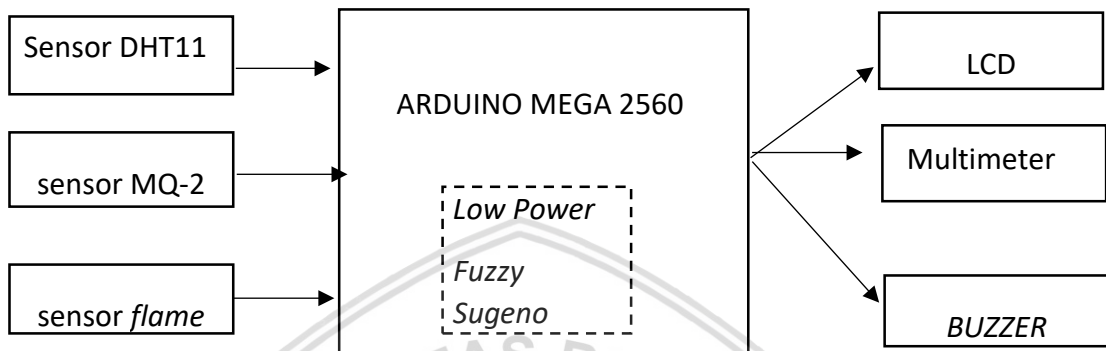
3.2 Rekayasa Kebutuhan

Rekayasa kebutuhan bertujuan untuk mendapatkan seluruh kebutuhan yang diperlukan dalam perancangan dan pengujian sistem. Rekayasa kebutuhan dilakukan dengan cara melakukan identifikasi kebutuhan system, serta peralatan yang terlibat didalamnya. Dalam kebutuhan sistem terjadi proses identifikasi terhadap beberapa perangkat yang digunakan seperti perangkat keras maupun

perangkat lunak. Dengan adanya proses identifikasi tersebut dapat memudahkan pengimplementasian sistem.

3.3 Perancangan

Pada tahap perancangan menjelaskan tentang perancangan sistem yang di tunjukkan pada Gambar 3.2.



Gambar 3.2 Diagram Blok Sistem

Pada perancangan ini terdapat 3 blok sensor sebagai input, yaitu sensor DHT11 sebagai pembaca suhu, sensor MQ-2 sebagai pendeteksi asap, dan sensor *flame* sebagai pendeteksi api. Ketiga sensor akan mengirimkan pembacaan dan diterima oleh Arduino Mega sebagai mikrokontroler untuk diolah. Pada Arduino Mega terdapat metode *low power* yang berarti jika tidak ada input dari sensor maka mikrokontroler akan berada pada mode *sleep* namun jika terdapat input dari sensor maka mikrokontroler berada pada mode *wake*. Ketika pada mode *wake* mikrokontroler akan menampilkan pembacaan sensor melalui LCD dan memberikan peringatan melalui *Buzzer*. Pada rancangan sistem ini multimeter berfungsi sebagai pembanding tegangan ketika sistem berada pada kondisi *wake* dan ketika sistem berada pada kondisi *sleep*.

3.4 Implementasi

Implementasi dilakukan berdasarkan perancangan sistem yang dijelaskan pada Gambar 3.2. langkah yang harus dilakukan yaitu:

3.4.1 Implementasi Perangkat Keras

Pada tahap konfigurasi perangkat keras, yang harus dilakukan adalah dengan merancang sensor MQ-2, sensor DHT11, dan sensor *flame* untuk disambungkan ke perangkat Arduino Mega. Sensor-sensor akan disambungkan melalui media kabel jumper male to female dan male to male. Konfigurasi perangkat keras dilakukan berdasarkan datasheet masing-masing sensor yang digunakan dan Arduino MEGA.

3.4.2 Implementasi Perangkat Lunak

Pada tahap konfigurasi perangkat lunak, dilakukan penginstallan software Arduino IDE yang dapat diunduh melalui website resmi dari Arduino. Setelah

itu, akan dilakukan coding program dengan menggunakan Bahasa C untuk mengkonfigurasi jalannya sistem pendeteksi kondisi kebakaran dan juga metode *low power* untuk menentukan output berdasarkan perhitungannya.

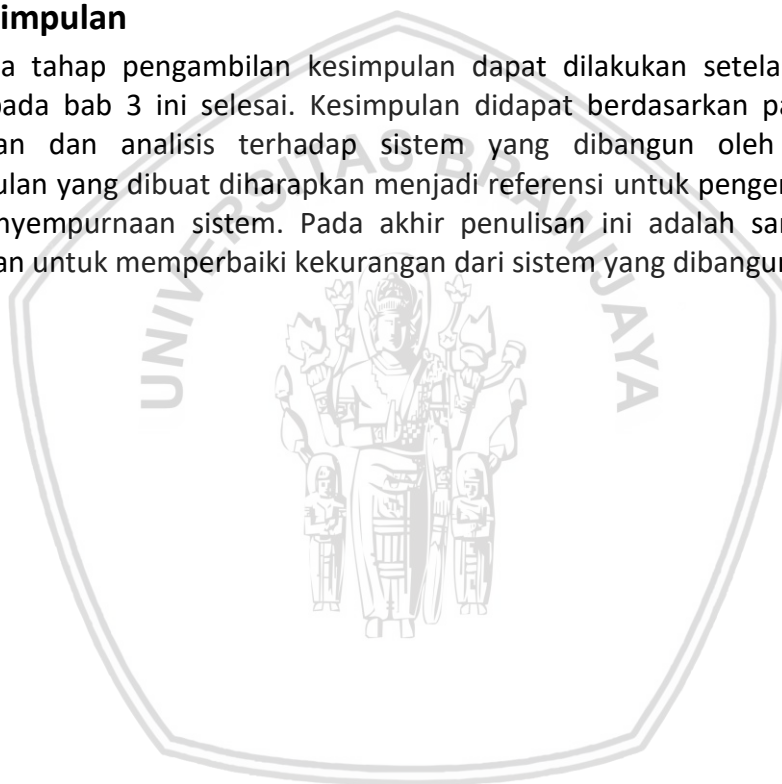
3.5 Pengujian dan Analisis

Pengujian akan dilakukan kepada seluruh sistem yaitu dengan menampilkan hasil pengujian terhadap output yang telah diberikan meliputi:

1. Pengujian hasil sensing setiap modul sensor yang digunakan untuk mendeteksi kebakaran
2. Pengujian perubahan nilai konsumsi arus ketika menggunakan metode *low power* dan tidak menggunakan *low power*

3.6 Kesimpulan

Pada tahap pengambilan kesimpulan dapat dilakukan setelah semua tahap pada bab 3 ini selesai. Kesimpulan didapat berdasarkan pada hasil pengujian dan analisis terhadap sistem yang dibangun oleh penulis. Kesimpulan yang dibuat diharapkan menjadi referensi untuk pengembangan dan penyempurnaan sistem. Pada akhir penulisan ini adalah saran yang bertujuan untuk memperbaiki kekurangan dari sistem yang dibangun.



BAB 4 REKAYASA KEBUTUHAN

Pada bab ini dijelaskan mengenai kebutuhan-kebutuhan yang harus dipenuhi untuk perancangan dan implementasi sistem. Rekayasa kebutuhan diharapkan sistem yang akan dibangun dapat bekerja dengan baik.

4.1 Gambaran Umum Sistem

Sistem pendeteksi kebakaran pada rumah menggunakan metode fuzzy sugeno dan *low power* merupakan sebuah alat pendeteksi kebakaran yang dapat meminimalisir banyak kerugian korban jiwa dalam bencana kebakaran dan juga ikut berperan dalam upaya penghematan listrik. Sistem dapat dikatakan berhasil jika bisa menentukan keadaan ruangan dalam kondisi tidak kebakaran, berasap, berapi asap sedikit, dan berapi asap banyak berdasarkan dari asap, suhu dan api yang terdeteksi oleh sensor kemudian data yang terbaca dari sensor akan diolah oleh mikrokontroler Arduino Mega 2560 dimana logika *Fuzzy Sugeno* sebagai penentu keputusan output. Namun apabila sistem tidak mendeteksi adanya kebakaran maka sistem akan berada pada mode *low power* dimana mikrokontroler pada mode sleep.

4.2 Analisis Kebutuhan Sistem

Analisis kebutuhan sistem dilakukan agar seluruh kebutuhan yang diperlukan sistem pendeteksi kebakaran dan *low power* dapat dipenuhi. Dalam analisis kebutuhan sistem terbagi atas 2 bagian, yaitu kebutuhan fungsional dan kebutuhan non fungsional.

4.2.1 Kebutuhan Fungsional

Berikut ini merupakan kebutuhan fungsional dari sistem yang harus terpenuhi agar sistem dapat berjalan sesuai dengan yang diinginkan, yaitu :

1. Sistem dapat mendeteksi suhu, asap dan api didalam ruangan.
Sistem dapat membaca input dari sensor DHT11, sesor MQ-2, dan sensor *flame*. Sensor DHT11 akan membaca suhu ruangan dengan rentang mencapai 100°C. Sensor MQ-2 berfungsi untuk mendeteksi adanya asap didalam ruangan. Sensor *flame* berfungsi untuk mendeteksi adanya api pada ruangan, sensor ini mendeteksi adanya api didalam ruangan menggunakan pancaran inframerah yang ada pada api.
2. Sistem dapat menjalankan perhitungan *fuzzy* melalui input dari sensor DHT11, sensor MQ-2 dan sensor *flame*.
Fungsi ini berperan penting agar logika *fuzzy* dapat menghasilkan *output* yang sesuai. Jika fungsi ini tidak berjalan dengan baik akan menghasilkan output yang tidak sesuai dengan perhitungan *fuzzy*. Ketika sistem bekerja dan mendapatkan *input* dari ketiga sensor, maka sistem akan melakukan

perhitungan untuk menentukan nilai fungsi keanggotaan. Ketika sistem sudah mendapatkan fungsi keanggotaan, maka sistem akan melakukan pengecekan terhadap yang sesuai dengan nilai keanggotaan.

3. Sistem dapat menjalankan fitur *low power*.
Pada saat sistem tidak mendeteksi adanya keadaan bahaya maka sistem akan beralih kedalam mode *sleep*. Dengan memanfaatkan mode *sleep* daya yang digunakan dapat berkurang, karena mikrokontroller mematikan fungsi-fungsi yang tidak perlukan. Untuk beralih dari mode *sleep* dan berpindah pada mode *wake* diperlukan adanya fitur *wake from interrupt* dengan cara membaca perubahan pada pin *interrupt*. Dengan membaca adanya peningkatan status maka sistem akan terbangun dari mode *power down sleep*.
4. Sistem dapat menampilkan kondisi pada LCD dan peringatan melalui *buzzer* sesuai dengan perhitungan *fuzzy*.
Fungsi ini berguna untuk menampilkan kondisi dari perhitungan *fuzzy* pada LCD agar user dapat lebih mudah mengamati kondisi didalam ruangan berupa tidak ada kebakaran, berasap, berapi asap sedikit, dan berapi asap banyak. Buzzer berfungsi sebagai alarm apabila kondisi semakin berbahaya.

4.2.2 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional pada sistem akan dibagi menjadi 2 kebutuhan, yaitu kebutuhan perangkat keras dan kebutuhan perangkat lunak.

4.2.2.1 Kebutuhan Perangkat Keras

Dalam membuat sistem pendeteksi kebakaran dengan fitur *low power* dibutuhkan beberapa perangkat keras, yaitu :

1. Arduino Mega 2560
Arduino Mega 2560 merupakan mikrokontroller yang memiliki memori yang cukup besar untuk menjalankan perhitungan fuzzy jika dibandingkan dengan Arduino yang lainnya, dimana pada penelitian ini kebutuhan memori yang dibutuhkan sebesar 35kb. Maka dari itu Arduino Mega 2560 merupakan sebuah mikrokontroller yang berperan sebagai inti dari pemrosesan yang berfungsi untuk mengolah hasil pembacaan yang didapat dari sensor DHT11, sensor MQ-2, dan sensor *flame* untuk dilakukan perhitungan kedalam logika *fuzzy sugeno*. Pada Gambar 4.2 merupakan spesifikasi dari Arduino Mega 2560.



Gambar 4.1 Arduino Mega 2560

Sumber: (Arduino, 2009)

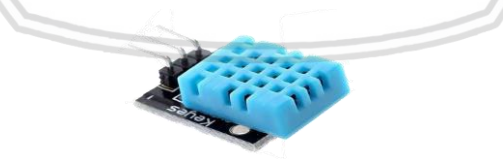
Deskripsi	Arduino UNO R3	Arduino Mega 2560
Chip mikrokontroler	ATmega328P	ATmega2560
Tegangan operasi	5V	
Tegangan input (rekendasi, via jack DC)	7V - 12V	7V - 12V
Tegangan input (limit, via jack DC)	6V - 20V	6V - 20V
Digital I/O pin	14 buah, 6 diantaranya PWM	54 buah, 6 diantaranya PWM
Analog Input pin	6 buah	16 buah
Arus DC per pin I/O	40 mA	20 mA
Arus DC pin 3.3V	50 mA	50 mA
Memori Flash	32 KB, 0.5 KB digunakan bootloader	256 KB, 8 KB digunakan bootloader
SRAM	2 KB	8 KB
EEPROM	1 KB	4 KB
Clock speed	16 Mhz	16 Mhz
Dimensi	68,6 mm x 53,4 mm	101,5 mm x 53,4 mm
Berat	25 g	37 g

Gambar 4.2 Spesifikasi Arduino Mega 2560 dan Arduino Uno

Sumber : (Arduino, 2009)

2. Sensor DHT11

Sensor DHT11 dibutuhkan karena punya stabilitas yang baik serta fitur kalibrasi yang akurat. Sensor DHT11 memiliki library khusus yang bernama DHT library, library tersebut berguna untuk memudahkan pengguna pemrogram di mikrokontroler (Akhmalia, 2017). Sensor DHT11 jika dibandingkan dengan sensor LM35 memiliki tingkat error yang lebih kecil. Gambar 4.3 merupakan gambar dari sensor DHT11. Berikut merupakan rincian lebih lengkap mengenai spesifikasi dari Sensor DHT11 pada Tabel 4.1



Gambar 4.3 Sensor DHT11

Sumber: (MySensors, 2017).

Tabel 4.1 Spesifikasi Sensor DHT11

<i>Supply Voltage</i>	3-5 Volt
<i>Temperature range</i>	0-50 °C Error ± 2 °C
<i>humidity</i>	20-90% RH Error ± 5 % RH

Sumber: (D-Robotics UK, 2010)

3. Sensor MQ-2

Sensor MQ-2 dibutuhkan karena sensor yang dapat mendeteksi H₂, LPG, CH₄, CO, Alkohol, dan Asap. Kelebihan dari Sensor MQ-2 yaitu menggunakan pembacaan melalui pembacaan ADC yang akan dikalibrasi menjadi nilai ppm(*part per million*). Sensor MQ-2 digunakan sebagai *input* asap kedalam sistem. Sensitivitas dari sensor dapat diatur melalui potensiometer.



Gambar 4.4 Sensor MQ-2

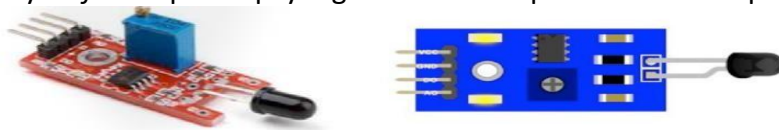
Sumber: (3D, 2014)

Item	Parameter	Min	Typical	Max	Unit
VCC	Working Voltage	4.9	5	5.1	V
PH	Heating consumption	0.5	-	800	mW
RL	Load resistance	-	adjustable	-	
RH	Heater resistance	-	33	-	Ω
Rs	Sensing Resistance	3	-	30	kΩ

Gambar 4.5 Spesifikasi Sensor MQ-2

4. Sensor *flame*

Sensor *flame* dibutuhkan karena modul ini sensitif terhadap api dan spektrum api. Cara kerjanya hanya mendeteksi panjang gelombang cahaya antara 760nm-1100nm yang merupakan ukuran untuk cahaya inframerah. Namun, jangkauan deteksi yang hampir 3 kaki dan deteksi sudut 60 yang tidak sangat besar untuk memantau satu kamar. Berfungsi mendeteksi adanya api didalam ruangan dimana sensor flame mendeteksi adanya *infrared* pada api yang memiliki nilai pembacaan berupa tegangan.



Gambar 4.6 Sensor *Flame*

Sumber: (Electronics, 2012)

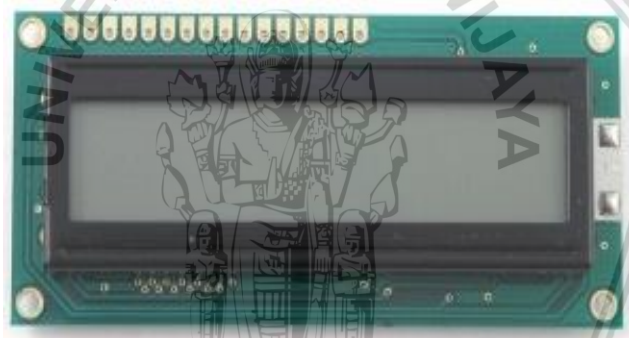
Tabel 4.2 Spesifikasi Sensor Flame

VCC	3.3-5V
Detection Distance	20 – 100 cm
Detection Angle	$\pm 60^\circ$
A0	Analog Output
D0	Digital Output

5. LCD

LCD (Liquid Cristal Display) dibutuhkan untuk menampilkan karakter angka, huruf ataupun simbol dengan lebih baik dan dengan konsumsi arus yang rendah.

Pada penelitian ini LCD berfungsi menampilkan kondisi ruangan yang terdiri dari 4 kondisi, tidak ada kebakaran, berasap, berapi asap sedikit, dan berapi asap banyak. LCD 2x16 dibutuhkan karena kebutuhan tampilan output membutuhkan banyaknya karakter yang ditampilkan pada LCD.



Gambar 4.7 LCD 2x16

Sumber: (BasicX, 2005)

6. Buzzer

Buzzer dibutuhkan karena sebagai output peringatan apabila sistem membaca ruangan dalam keadaan bahaya, yaitu berasap, berapi asap sedikit dan berapi asap banyak.

Tabel 4.3 Deskripsi Pin Buzzer

Simbol	Deskripsi
VCC (+)	Voltage 3,3-5v
GND	Ground



Gambar 4.8 Buzzer

Sumber: (Co, 2015)

7. Laptop
Untuk melakukan pemrograman pada Arduino Mega 2560 melalui Arduino IDE dan memonitoring hasil input melalui serial monitor pada Arduino IDE.

4.2.2.2 Kebutuhan perangkat Lunak

Kebutuhan perangkat lunak yang dibutuhkan dalam sistem ini, yaitu :

1. Library dht.h
Library dht.h merupakan sebuah program library yang diperuntukkan agar sensor DHT11 dapat berjalan secara maksimal dalam membaca suhu ruangan. Didalam library dht.h terdapat perhitungan untuk menampilkan pembacaan suhu dalam satuan Celcius.
2. Library LiquidCrystal.
Library yang digunakan untuk melakukan inisialisasi terhadap LCD 2x16 baik dari penggunaan cursor dan backlight.
3. Library wire.h
Merupakan library yang diperlukan apabila LCD menggunakan modul I2C.
4. Fuzzy
Sebagai penentu keputusan output yang sesuai dengan perhitungan *fuzzy sugeno*.

4.3 Batasan Desain Sistem

Dalam pembuatan prototipe sistem pendeteksi kebakaran ini terdapat beberapa batasan yang harus diterapkan sehingga dalam proses pengerjaan dapat lebih terfokus. Batasan-batasan dari desain sistem tersebut adalah:

1. Sistem membaca kondisi ruangan melalui 3 buah sensor, yaitu sensor DHT11, sensor MQ-2, sensor *flame*.
2. Pembacaan dari ketiga sensor diolah melalui perhitungan logika *fuzzy sugeno*.
3. Program *fuzzy* yang digunakan dalam sistem harus sesuai dengan perhitungan dan *rule*.
4. Sistem akan berada pada mode *sleep* apabila sistem tidak mendeteksi adanya peningkatan status.
5. Sistem akan berpindah dari mode *sleep* ke mode *wake* apabila ada peningkatan status di dalam ruangan.

BAB 5 PERANCANGAN DAN IMPLEMENTASI

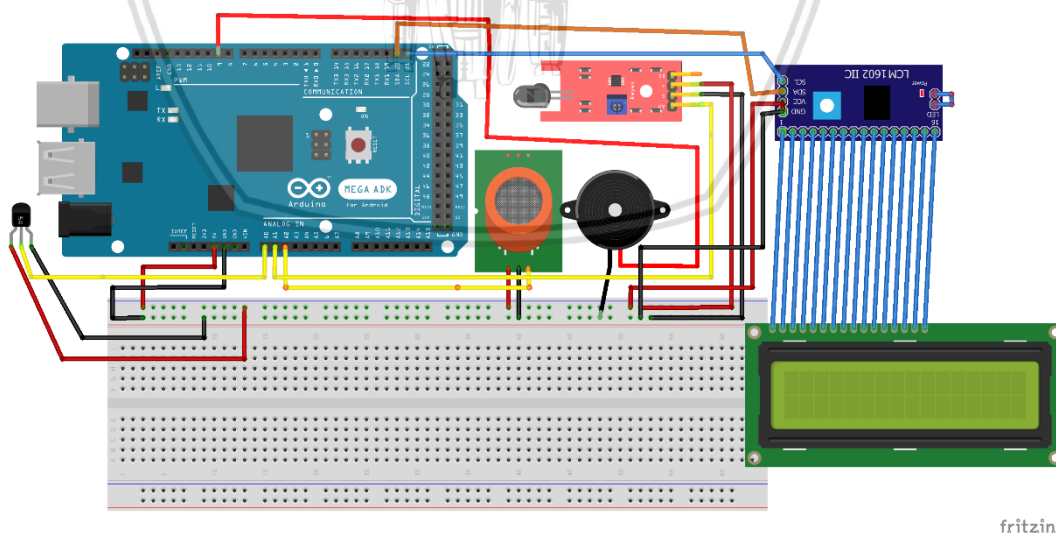
Pada bab ini akan menjelaskan mengenai perancangan dan implementasi dari sistem pendeteksi kebakaran dalam ruangan dengan menggunakan metode *low power down sleep* dengan memanfaatkan sensor *flame* sebagai *interrupt* untuk kembali pada mode *wake*. Dalam bab ini perancangan terbagi atas perancangan perangkat keras dan perancangan perangkat lunak dari sistem dan implementasi perangkat keras dan perangkat lunak dari sistem yang telah dirancang.

5.1 Perancangan Sistem

Perangkat yang akan dirancang dalam penelitian ini adalah sebuah sistem yang dapat mendeteksi kebakaran berdasarkan suhu yang ada didalam ruangan, intensitas asap, dan keberadaan api didalam ruangan serta dapat menjalankan kondisi *sleep power down* ketika tidak ada indikasi kebakaran dalam ruangan. Input dalam sistem ini berasal dari suhu, asap, dan api yang kemudian akan diolah menggunakan *mikrokontroler* yang didalamnya terdapat logika *fuzzy* dan *low power*. Output pada sistem ini berupa kondisi hasil dari perhitungan logika *fuzzy*.

5.1.1 Perancangan Perangkat Keras

Perangkat keras diperlukan untuk membangun sistem agar dapat menerapkan logika *fuzzy* sebagai output dan mekanisme *low power* pada sistem pendeteksi kebakaran. Mikrokontroler yang digunakan untuk memproses data *input* adalah *Arduino Mega 2560*. Input yang digunakan untuk di proses adalah input suhu dengan menggunakan sensor DHT11, input asap dengan menggunakan sensor MQ-2, data input api dengan menggunakan sensor *flame*.



Gambar 5.1 Skema Perancangan Perangkat Keras

Tabel 5.1 Koneksi Pin Perancangan Perangkat Keras

Pin Arduino Mega	Pin MQ-2	Pin DHT11	Pin <i>Flame</i>	Pin <i>Buzzer</i>	Pin LCD
Vcc	VCC	VCC	VCC		VCC
GND	GND	GND	GND	GND	GND
A0		A0			
A1			A0		
A2	A0				
9				IN	
SDA					SDA
SCL					SCL

Keterangan :

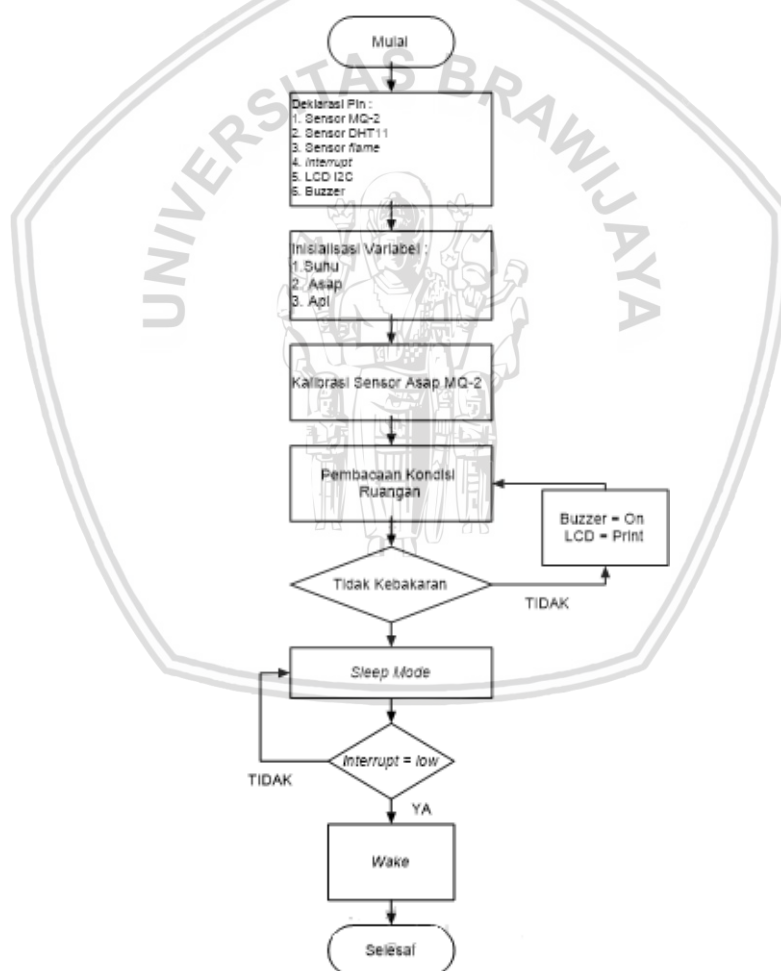
- Pin VCC Arduino Mega 2560 dihubungkan kepada seluruh pin VCC pada setiap sensor yang akan dihubungkan, yakni sensor MQ-2, sensor DHT11, sensor *flame* dan LCD 2x16. Karena Pin VCC merupakan sumber arus yang dibutuhkan untuk menjalankan pembacaan oleh sensor.
- Pin Ground(GND) Arduino Mega 2560 dihubungkan kepada seluruh pin GND yang ada pada setiap modul/ sensor yang akan dihubungkan, yakni sensor MQ-, sensor DHT11, sensor *flame*, *buzzer*, dan LCD 2x16.
- Pin Analog(A0) pada sensor MQ-2 dihubungkan pada pin Analag 2 (A2) yang ada pada Arduino Mega 2560. Karena sensor MQ-2 merupakan yang dapat berjalan pada pin Analog.
- Pin Analog(A0) pada sensor DHT11 dihubungkan pada pin Analog 0 (A0) yang ada pada Arduino Mega 2560.
- Pin Analog(A0) pada sensor *flame* dihubungkan pada pin Analog 1 (A1) yang ada pada Arduino Mega2560.
- Buzzer (+) dihubungkan pada pin Digital 9 pada Arduino Mega 2560. Karena buzzer hanya memiliki logika HIGH dan LOW maka buzzer dihubungkan pada pin digital.
- Pin SDA dan SCL yang terdapat pada modul I2C dihubungkan pada pin SDA dan SCL yang ada pada Arduino Mega 2560.

5.1.2 Perancangan Perangkat Lunak

Perangkat lunak yang digunakan pada sistem ini berupa *code* program yang dibuat dalam *software mikrokontroller* yang telah disediakan yaitu Arduino IDE. Program yang dibuat yaitu dengan menerapkan logika *fuzzy* sebagai pengambil keputusan output berdasarkan 3 input yaitu dari sensor DHT11 untuk mengetahui

kondisi suhu ruangan, sensor MQ-2 untuk mengetahui intensitas asap didalam ruangan, dan sensor *flame* untuk mengetahui keberadaan api didalam ruangan. Serta program *low power* sebagai perubahan *state* apabila sistem tidak mendeteksi adanya keberadaan api didalam ruangan. Data input dari 3 sensor akan disimpan dalam bentuk variabel yang akan di proses menggunakan *fuzzy* untuk menentukan output melalui himpunan *fuzzy* dari masing-masing variabel.

Dalam implementasi *fuzzy* akan dibuat suatu variabel array yang berfungsi untuk menyimpan hasil fuzzyfikasi dari setiap variabel yaitu, suhu, asap, dan api. Selain variabel untuk fuzzyfikasi, terdapat variabel untuk inferensi logika *fuzzy* dan defuzzifikasi. Inferensi logika *fuzzy* dilakukan dengan menggunakan metode *min* untuk menentukan output dari tiap rule. Sedangkan variabel defuzzifikasi menggunakan metode *max* untuk perhitungan terhadap himpunan tegas (*crisp*) sehingga didapatkan *output* dari sistem. Output dari sistem ini merupakan *buzzer* dan LCD berfungsi menampilkan kondisi hasil dari defuzzifikasi.



Gambar 5.2 Flowchart perancangan perangkat lunak sistem

5.1.2.1 ADC

- Sensor Asap

Sensor asap memiliki range pembacaan mulai dari 0-5 volt yang memiliki range pembacaan sampai 1000ppm. Dimana pada saat pembacaan tidak ada asap analog read membaca sebesar 24. maka perhitungannya dapat dilihat sebagai berikut :

Jangkauan Tegangan : volt maksimum – volt minimum

$$: 5v - 0v = 5volt$$

$$\text{Tegangan terbaca} : 24 \times \frac{5 \text{ volt}}{1023} = 0,11736 \text{ volt}$$

$$\text{ADC minimum} : \frac{0 \text{ volt}}{1023} = 0$$

$$\text{ADC maksimum} : \frac{5 \text{ volt}}{1023} = 0,00489$$

Maka dapat dikatakan pembacaan analog pada angka 24 yang merupakan tegangan terbaca direpresentasikan dengan 0,11736 volt

- Sensor *Flame*

Sensor flame memiliki range pembacaan mulai dari 3,5 – 5,5 volt. Dimana ketika Sensor flame tidak mendeteksi adanya api analog read membaca sebesar 1021. Berikut adalah perhitungan Vref untuk sensor flame:

Jangkauan Tegangan : volt maksimum – volt minimum

$$: 5,5v - 3,5v = 2volt$$

$$1021 \times \frac{2 \text{ volt}}{1023} = 2,00508 \text{ volt} + 3,5 \text{ volt} = 5,50508 \text{ volt}$$

$$\text{ADC Minimum} : \frac{3,5 \text{ volt}}{1023} = 0,00342 \text{ volt}$$

$$\text{ADC Maksimum} : \frac{5,5 \text{ volt}}{1023} = 0,00538 \text{ volt}$$

Maka dapat dikatakan pembacaan analog pada angka 1021 direpresentasikan dengan 2,00508 volt

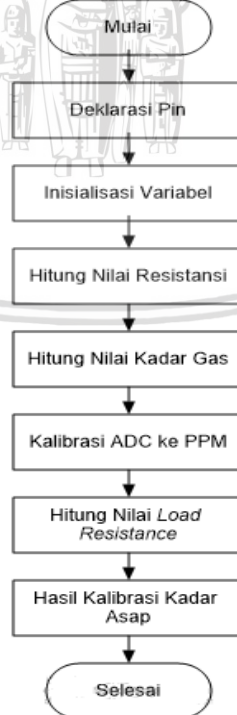
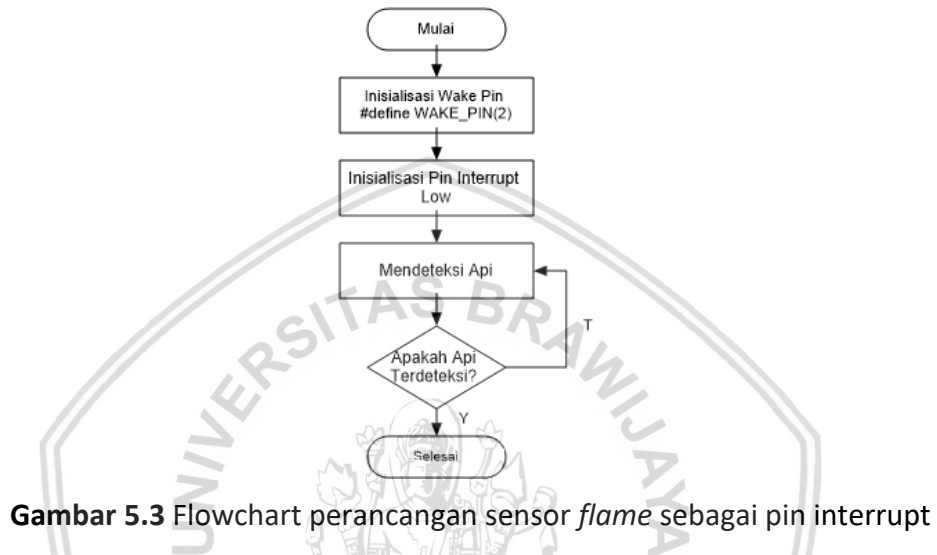
5.1.2.2 Perancangan perangkat lunak *sensor flame* sebagai *interrupt*

Interrupt pada sistem ini menggunakan sensor *flame* untuk membuat sistem berpindah dari *state SLEEP* menjadi *WAKE*. Pengaturan pin *interrupt* yang digunakan yaitu dengan memanfaatkan pembacaan *digital* dari sensor *flame*. Berikut merupakan flowchart program untuk *Interrupt* sensor *flame* dapat dilihat pada Gambar 5.3.

Pada inialisasi pin yang digunakan sebagai *WAKE_PIN* yaitu menggunakan pin Digital 2 (D2) yang ada pada Arduino Mega 2560. Pin *interrupt* diatur untuk memiliki logika awal LOW karena sensor *flame* akan berada pada pembacaan HIGH apabila mendeteksi adanya api. Apabila sensor *flame* berada pada pembacaan HIGH maka *interrupt* pun akan menjadi HIGH dan *mikrokontroler* dapat terbangun dari *mode sleep power down*.

5.1.2.3 Perancangan Perangkat Lunak Sensor Asap MQ-2

Perancangan perangkat lunak pada sensor asap digunakan untuk membaca nilai data kalibrasi yang dilakukan oleh sensor asap sehingga dapat diketahui kadar asap yang terbaca. Pada pembacaan sensor asap dibutuhkan waktu untuk melakukan pengkalibrasian dahulu untuk memuat hasil nilai data kadar asap. Sensor asap berjalan pada keadaan *wake* dan dijalankan ketika terjadi *interrupt* pada sistem. Berikut gambar 5.4 yang menampilkan perancangan perangkat lunak pada sensor gas MQ-2.



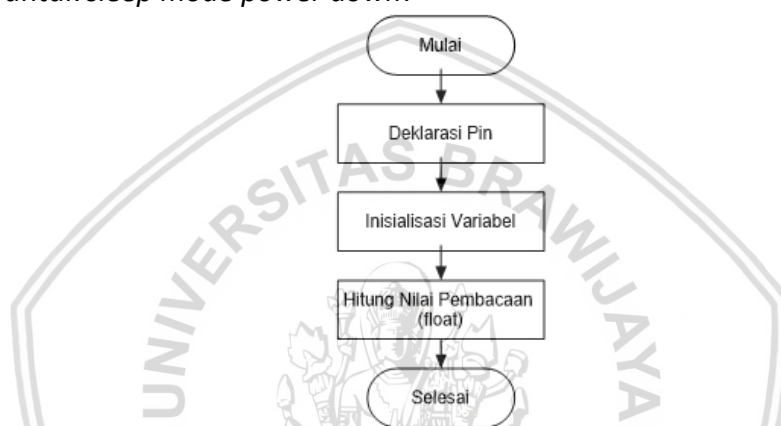
Gambar 5.4 Flowchart perancangan perangkat lunak sensor asap MQ-2

5.1.2.4 Perancangan perangkat lunak sensor suhu DHT11

Perancangan perangkat lunak pada sensor suhu DHT11 digunakan untuk membaca nilai data yang dilakukan sensor suhu untuk mengetahui suhu ruangan yang terbaca. Gambar 5.5 yang merupakan proses perancangan perangkat lunak sensor suhu DHT11.

5.1.2.5 Perancangan Perangkat Lunak *Sleep Mode Power Down*

Perancangan perangkat lunak sleep mode power down merupakan tujuan utama dari penelitian ini. Sebagai wujud penghematan daya listrik yang ada. Ketika sistem dalam kondisi *sleep* maka *sleep mode power down* dijalankan. Penggunaan *sleep mode power down* berjalan dengan mematikan beberapa fungsional sistem lain. Penjelasan lebih lanjut dapat dilihat pada Gambar 5.6 dalam aliran bagan diagram untuk *sleep mode power down*.



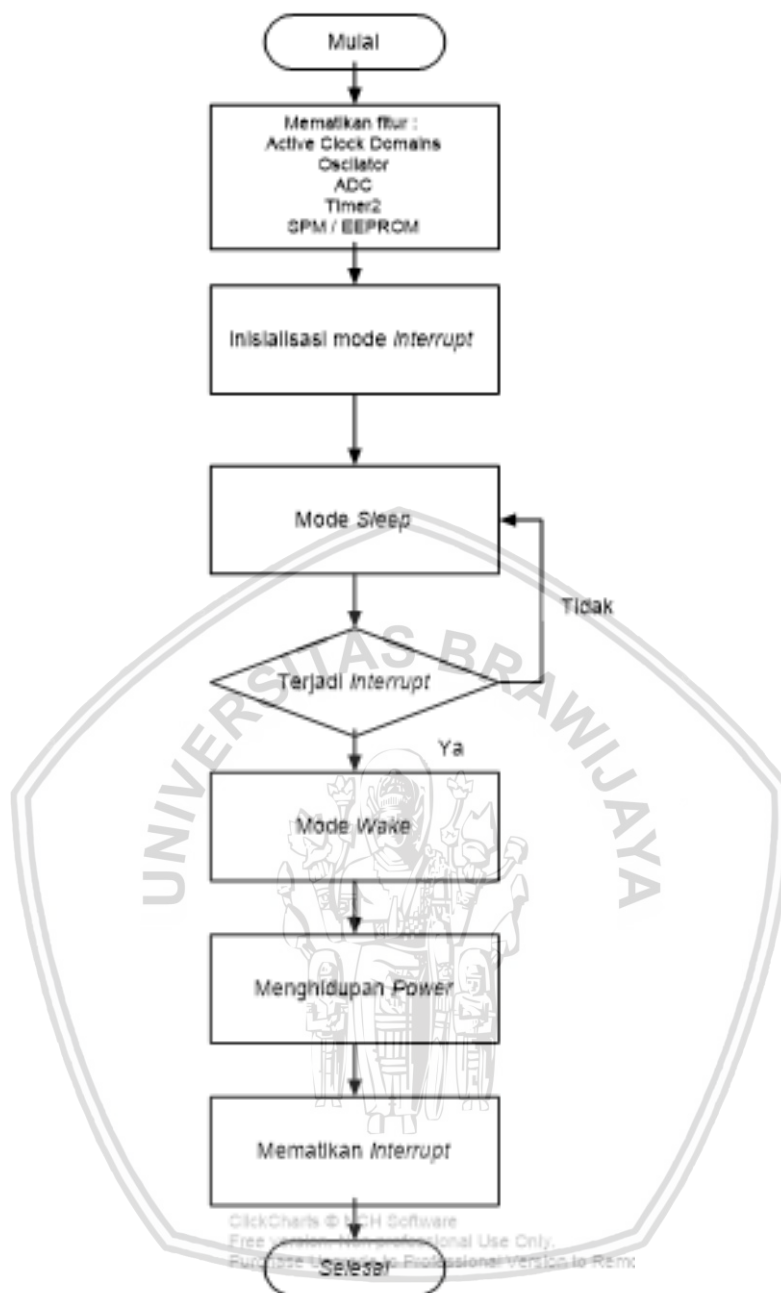
Gambar 5.5 Flowchart Perangkat Lunak Sensor Suhu

5.1.2.6 Perancangan Fuzzy

Perancangan sistem dengan menggunakan metode fuzzy membutuhkan beberapa proses sehingga terbentuknya suatu keputusan output dari sistem sesuai dengan perhitungan fuzzy. Proses tersebut antara lain fuzzifikasi, pembuatan rule, inferensi, dan defuzzifikasi. Pada Gambar 5.7 menunjukkan bahwa sub-proses pada control fuzzy memiliki fungsi yang saling berhubungan dengan sub-proses yang lain sehingga sub-proses yang dihasilkan akan menjadi input dari sub-proses berikutnya sampai menjadi output akhir dari sistem.

5.1.2.7 Perancangan Fuzzy

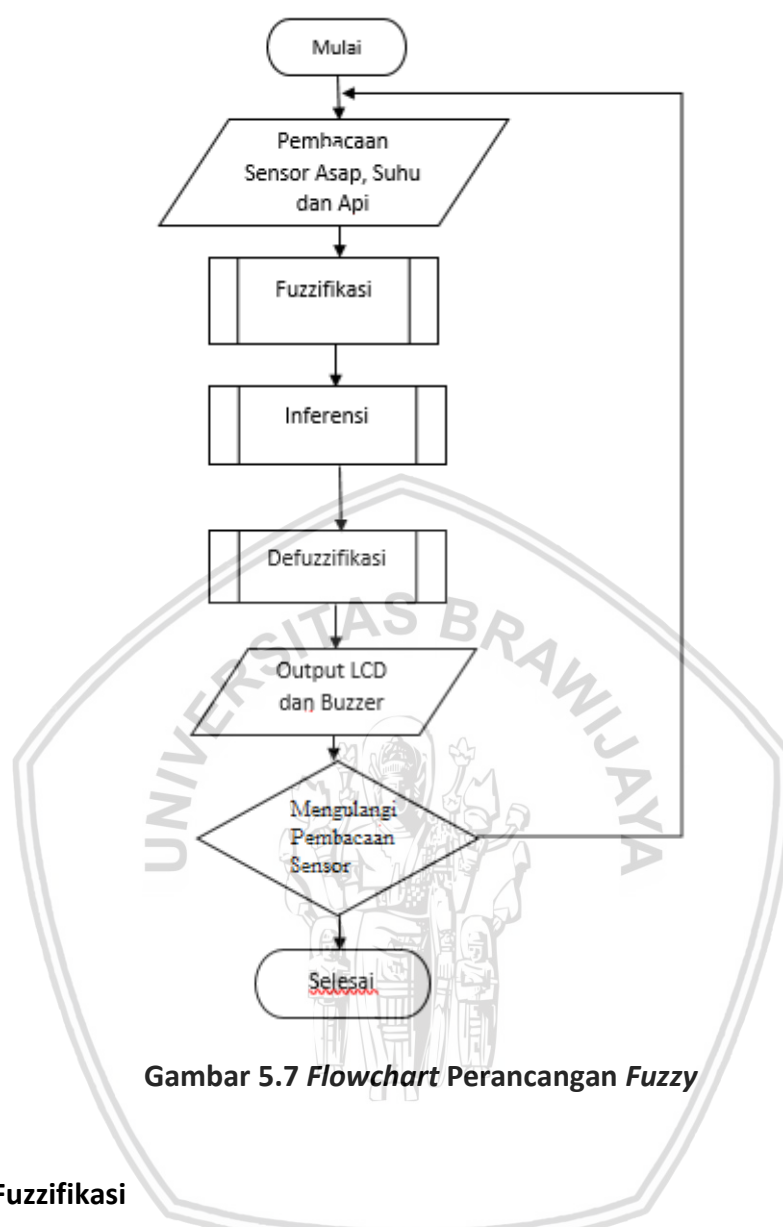
Perancangan sistem dengan menggunakan metode fuzzy membutuhkan beberapa proses sehingga terbentuknya suatu keputusan output dari sistem sesuai dengan perhitungan fuzzy. Proses tersebut antara lain fuzzifikasi, pembuatan rule, inferensi, dan defuzzifikasi. Pada Gambar 5.7 menunjukkan bahwa sub-proses pada control fuzzy memiliki fungsi yang saling berhubungan dengan sub-proses yang lain sehingga sub-proses yang dihasilkan akan menjadi input dari sub-proses berikutnya sampai menjadi output akhir dari sistem.



Gambar 5.6 Flowchart perancangan perangkat lunak *sleep mode power down*

5.1.2.8 Perancangan Fuzzy

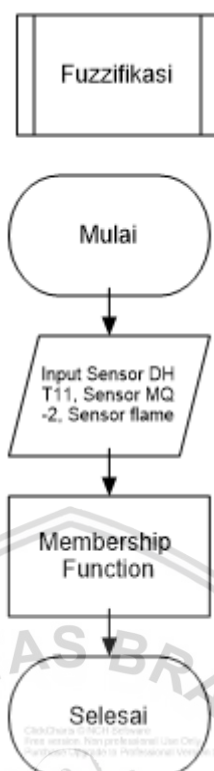
Perancangan sistem dengan menggunakan metode fuzzy membutuhkan beberapa proses sehingga terbentuknya suatu keputusan output dari sistem sesuai dengan perhitungan fuzzy. Proses tersebut antara lain fuzzifikasi, pembuatan rule, inferensi, dan defuzzifikasi. Pada Gambar 5.7 menunjukkan bahwa sub-proses pada control fuzzy memiliki fungsi yang saling berhubungan dengan sub-proses yang lain sehingga sub-proses yang dihasilkan akan menjadi input dari sub-proses berikutnya sampai menjadi output akhir dari sistem.



Gambar 5.7 Flowchart Perancangan Fuzzy

5.1.2.9 Fuzzifikasi

Fuzzifikasi adalah suatu pengubahan nilai tegas/*real* kedalam fungsi keanggotaan *fuzzy*. Gambar 5.8, sub-proses fuzzifikasi akan menerima proses *input* dari sensor DHT11 yang berfungsi untuk *input* data suhu, sensor MQ-2 yang berfungsi untuk *input* data asap, dan sensor *flame* yang berfungsi untuk *input* data api. Fungsi dari sub-proses fuzzifikasi adalah merubah nilai tegas ke dalam fungsi keanggotaan (*membership function*). Berikut ini adalah flowchart proses fuzzifikasi.



Gambar 5.8 Flowchart Proses Fuzzifikasi

5.1.2.10 Variabel Suhu

Pada variabel suhu, penulis akan menggunakan 4 himpunan fuzzy yaitu dingin, sejuk, hangat, dan panas berdasarkan (Nurromianto, 2015). Berikut ini merupakan himpunan pada variabel suhu.

- Dingin [0 20]

Fungsi keanggotaan dapat dirumuskan sebagai berikut.

$$\mu[Dingin] = \begin{cases} 1, & x \leq 10 \\ \frac{20-x}{20-10}, & 10 < x \leq 20 \\ 0, & x > 20 \end{cases} \quad (5.1)$$

- Sejuk [10 30]

Fungsi keanggotaan dapat dirumuskan sebagai berikut.

$$\mu[Sejuk] = \begin{cases} 0, & x \leq 10 \text{ atau } \geq 30 \\ \frac{x-10}{20-10}, & 10 < x \leq 20 \\ \frac{30-x}{30-20}, & 20 < x \leq 30 \\ 1, & x = 20 \end{cases} \quad (5.2)$$

- Hangat [20 40]

Fungsi keanggotaan dapat dirumuskan sebagai berikut.

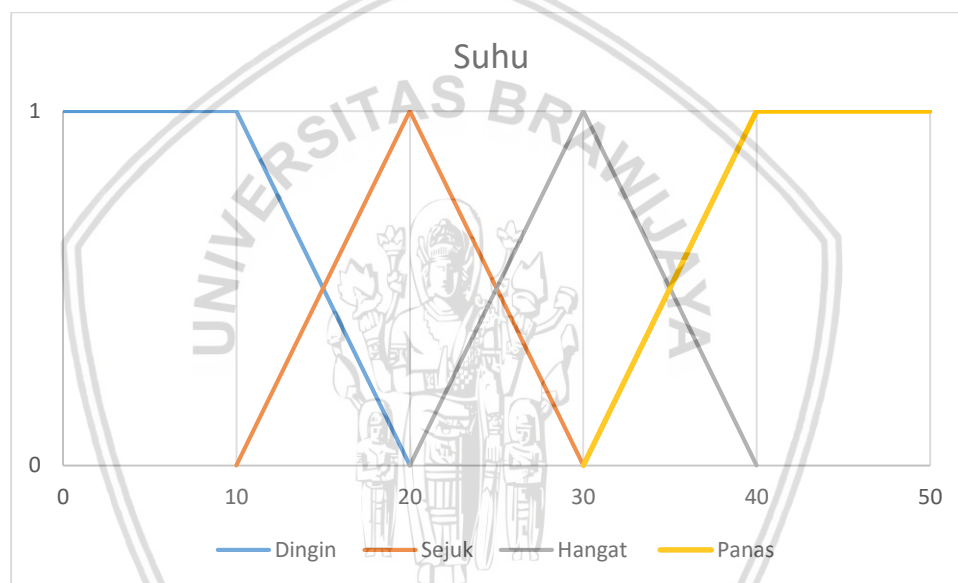
$$\mu[Hangat] = \begin{cases} 0, & x \leq 20 \text{ atau } \geq 40 \\ \frac{x-20}{30-20}, & 20 < x \leq 30 \\ \frac{40-x}{40-30}, & 30 < x \leq 40 \\ 1, & x = 30 \end{cases} \quad (5.3)$$

- Panas [>30]

Fungsi keanggotaan dapat dirumuskan sebagai berikut.

$$\mu[Panas] = \begin{cases} 1, & x \geq 40 \\ \frac{x-30}{40-30}, & 30 < x \leq 40 \\ 0, & x \leq 30 \end{cases} \quad (5.4)$$

Fungsi keanggotaan dari variabel suhu dapat direpresentasikan seperti Gambar 5.5.



Gambar 5.9 Membership Function Suhu

5.1.2.11 Variabel Asap

Pada pembacaan *input* asap menggunakan nilai keluaran berupa *ppm* (Part Per Million). Kebakaran terjadi biasanya diketahui dengan kenaikan asap pada udara normal yaitu lebih besar dari nilai 400 *ppm* (Prayogi, et al., 2016). Berikut merupakan himpunan keanggotaan dari variabel asap berdasarkan pada (Transmigrasi, 2011).

- Renggang [$0 \ 50$]

Berikut rumus fungsi keanggotaan dari variabel asap.

$$\mu[Renggang] = \begin{cases} 1, & x \leq 25 \\ \frac{50-x}{50-25}, & 25 < x \leq 50 \\ 0, & > 50 \end{cases} \quad (5.5)$$

- Sedang [25 75]

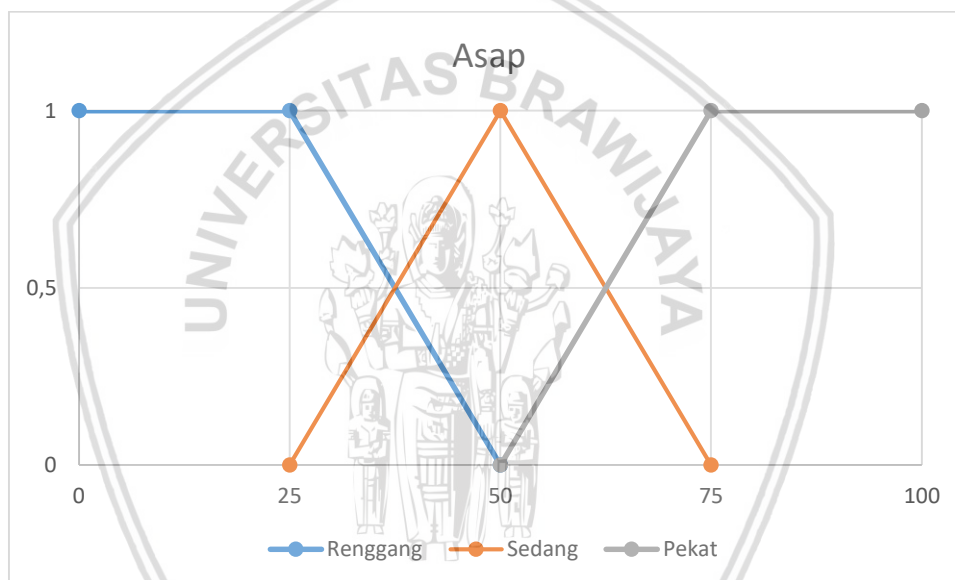
Berikut rumus fungsi keanggotaan dari variabel asap.

$$\mu[\text{Sedang}] = \begin{cases} 0, & x \leq 25 \text{ atau } > 75 \\ \frac{x-25}{50-25}, & 25 < x \leq 50 \\ \frac{75-x}{75-50}, & 50 < x \leq 75 \\ 1, & x = 50 \end{cases} \quad (5.6)$$

- Pekat [>50]

Berikut rumus fungsi keanggotaan dari variabel asap.

$$\mu[\text{Pekat}] = \begin{cases} 1, & x > 75 \\ \frac{x-50}{75-50}, & 50 < x \leq 75 \\ 0, & < 50 \end{cases} \quad (5.7)$$



Gambar 5.10 Membership Function Asap

5.1.2.12 Variabel Api

Pada pembacaan input api menggunakan pembacaan tegangan melalui serial monitor yang terdapat pada Arduino IDE. Menurut datasheet dari sensor *flame* pembacaan maksimal dari sensor ini merupakan 100 cm. Maka dilakukan percobaan dengan menyalakan api menggunakan korek api dalam 4 jarak yaitu api dekat sensor (≤ 25 cm), api agak dekat dengan sensor (50cm), dan api jauh dengan sensor (> 50 cm – 100 cm), dan tidak ada api. Berikut merupakan himpunan dari variabel api.

- Api dekat dengan Sensor [0 50]

Berikut rumus fungsi keanggotaan dari variabel api.

$$\mu[Dekat] = \begin{cases} 1, & x \leq 60 \\ \frac{100 - x}{100 - 60}, & 60 < x \leq 100 \\ 0, & x > 100 \end{cases}$$

- Api agak dekat dengan sensor [100 300]

Berikut rumus fungsi keanggotaan dari variabel api.

$$\mu[Agak Dekat] = \begin{cases} 0, & x \leq 60 \text{ atau } > 350 \\ \frac{x - 60}{100 - 60}, & 60 < x \leq 100 \\ \frac{350 - x}{350 - 300}, & 300 < x \leq 350 \\ 1, & 100 < x \leq 300 \end{cases}$$

- Jauh Dengan Sensor [350 600]

Berikut rumus fungsi keanggotaan dari variabel api.

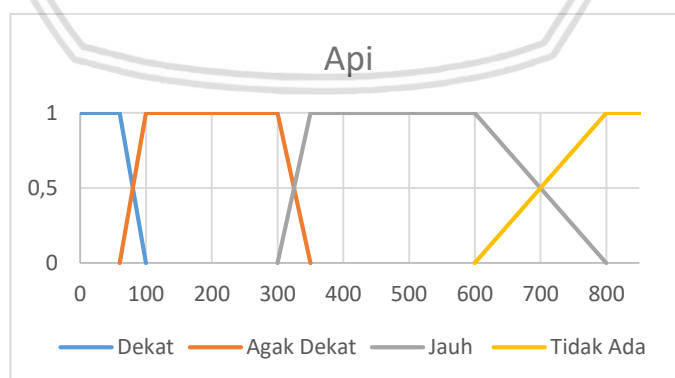
$$\mu[Jauh] = \begin{cases} 0, & x \leq 300 \text{ atau } > 800 \\ \frac{x - 300}{350 - 300}, & 300 < x \leq 350 \\ \frac{800 - x}{800 - 600}, & 600 < x \leq 800 \\ 1, & 350 < x \leq 600 \end{cases}$$

- Tidak ada api [>600]

Berikut rumus fungsi keanggotaan dari variabel api.

$$\mu[Tidak ada api] = \begin{cases} 1, & x > 800 \\ \frac{x - 600}{800 - 600}, & 600 < x \leq 800 \\ 0, & x < 600 \end{cases}$$

Berikut adalah gambar dari *membership function* dari variabel api.



Gambar 5.11 Membership Function Api

5.1.2.13 Rule

Rule based suatu bentuk aturan relasi/implikasi if-then. Fungsi implikasi adalah menyusun baris aturan berupa implikasi *fuzzy* yang menyatakan relasi antara variabel *input* dan *output*. Pada metode sugeno, fungsi implikasi yang digunakan adalah *min*. Pada pembuatan aturan, penulis akan membuat dengan menggunakan “IF” dan “AND” dan menghasilkan perintah “THEN”. Aturan fuzzy yang digunakan untuk menentukan kondisi *output* berupa normal / tidak ada kebakaran, berasap, berapi asap sedikit, dan berapi asap berdasarkan pada penelitian sebelumnya oleh (Purnomo, 2017) banyak dijelaskan pada tabel 5.2.

Tabel 5.2 Pembuatan Rule Fuzzy

Rule	Suhu	Asap	Api	Output
1	Dingin	Renggang	Dekat	Tidak Kebakaran
2	Dingin	Renggang	Agak dekat	Tidak Kebakaran
3	Dingin	Renggang	Jauh	Tidak Kebakaran
4	Dingin	Renggang	Tidak ada	Tidak Kebakaran
5	Dingin	Sedang	Dekat	Tidak Kebakaran
6	Dingin	Sedang	Agak dekat	Tidak Kebakaran
7	Dingin	Sedang	Jauh	Tidak Kebakaran
8	Dingin	Sedang	Tidak ada	Berasap
9	Dingin	Pekat	Dekat	Tidak Kebakaran
10	Dingin	Pekat	Agak dekat	Tidak Kebakaran
11	Dingin	Pekat	Jauh	Tidak Kebakaran
12	Dingin	Pekat	Tidak ada	Berasap
13	Sejuk	Renggang	Dekat	Tidak Kebakaran
14	Sejuk	Renggang	Agak dekat	Tidak Kebakaran
15	Sejuk	Renggang	Jauh	Tidak Kebakaran
16	Sejuk	Renggang	Tidak ada	Tidak Kebakaran
17	Sejuk	Sedang	Dekat	Tidak Kebakaran
18	Sejuk	Sedang	Agak dekat	Tidak Kebakaran
19	Sejuk	Sedang	Jauh	Tidak Kebakaran
20	Sejuk	Sedang	Tidak ada	Berasap
21	Sejuk	Pekat	Dekat	Tidak Kebakaran
22	Sejuk	Pekat	Agak dekat	Tidak Kebakaran
23	Sejuk	Pekat	Jauh	Tidak Kebakaran

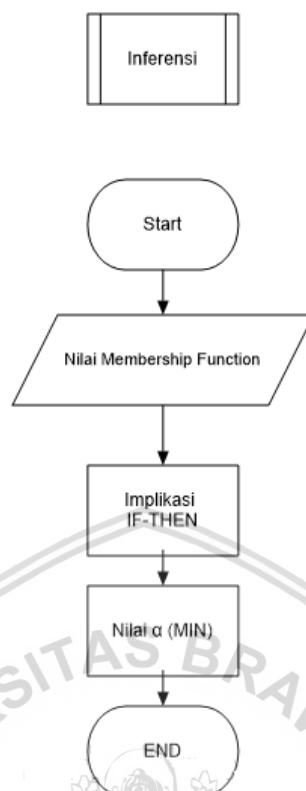
Rule	Suhu	Asap	Api	Output
24	Sejuk	Pekat	Tidak ada	Berasap
25	Hangat	Renggang	Dekat	Tidak Kebakaran
26	Hangat	Renggang	Agak dekat	Tidak Kebakaran
27	Hangat	Renggang	Jauh	Tidak Kebakaran
28	Hangat	Renggang	Tidak ada	Tidak Kebakaran
29	Hangat	Sedang	Dekat	Berapi asap sedikit
30	Hangat	Sedang	Agak dekat	Berapi asap sedikit
31	Hangat	Sedang	Jauh	Berapi asap sedikit
32	Hangat	Sedang	Tidak ada	Berasap
33	Hangat	Pekat	Dekat	Berapi asap banyak
34	Hangat	Pekat	Agak dekat	Berapi asap banyak
35	Hangat	Pekat	Jauh	Berapi asap banyak
36	Hangat	Pekat	Tidak ada	Berasap
37	Panas	Renggang	Dekat	Tidak Kebakaran
38	Panas	Renggang	Agak dekat	Tidak Kebakaran
39	Panas	Renggang	Jauh	Tidak Kebakaran
40	Panas	Renggang	Tidak ada	Tidak Kebakaran
41	Panas	Sedang	Dekat	Berapi asap sedikit
42	Panas	Sedang	Agak dekat	Berapi asap sedikit
43	Panas	Sedang	Jauh	Berapi asap sedikit
44	Panas	Sedang	Tidak ada	Berasap
45	Panas	Pekat	Dekat	Berapi asap banyak
46	Panas	Pekat	Agak dekat	Berapi asap banyak
47	Panas	Pekat	Jauh	Berapi asap banyak
48	Panas	Pekat	Tidak ada	Berasap

5.1.2.14 Inferensi

Inference engine merupakan suatu proses implikasi dalam menalar nilai input untuk menentukan nilai output sebagai bentuk pengambilan keputusan. Berdasarkan pembuatan rule pada Tabel 5.2 maka akan ditentukan nilai α pada setiap aturan. α merupakan nilai keanggotaan baru hasil operasi 2 himpunan atau lebih. Aturan/rule yang digunakan adalah MIN pada setiap fungsi implikasi. Sehingga rumus untuk mendapatkan inferensi adalah sebagai berikut

$$\mu A \cap B[x] = \min(\mu_A[x], \mu_B[x])$$

Berikut ini adalah flowchart proses inferensi sehingga didapatkan nilai α untuk mendapatkan nilai keanggotaan baru.



Gambar 5.12 Flowchart Proses Inferensi

Pada Gambar 5.12 dijelaskan bahwa inferensi membutuhkan input dari nilai membership function yang didapatkan pada proses sub-proses fuzzifikasi. nilai dari membership function akan dicocokkan dengan rule yang telah dibuat. Setelah didapatkan rule yang cocok, maka diterapkan metode min untuk mencari nilai terkecil dari setiap rule sehingga didapatkan nilai α yang merupakan nilai keanggotaan baru.

Berikut ini merupakan contoh dari penggunaan inferensi. Pada sebuah perhitungan fuzzifikasi didapatkan nilai keanggotaan suhu sejuk adalah 0,1, suhu hangat adalah 0,9, api jauh dengan sensor adalah 1, dan asap sedang adalah 1. Maka hasil nilai keanggotaan baru adalah sebagai berikut.

- **RULE[19] : IF SUHU SEJUK && API JAUH DENGAN SENSOR && ASAP SEDANG THEN **TIDAK ADA KEBAKARAN****

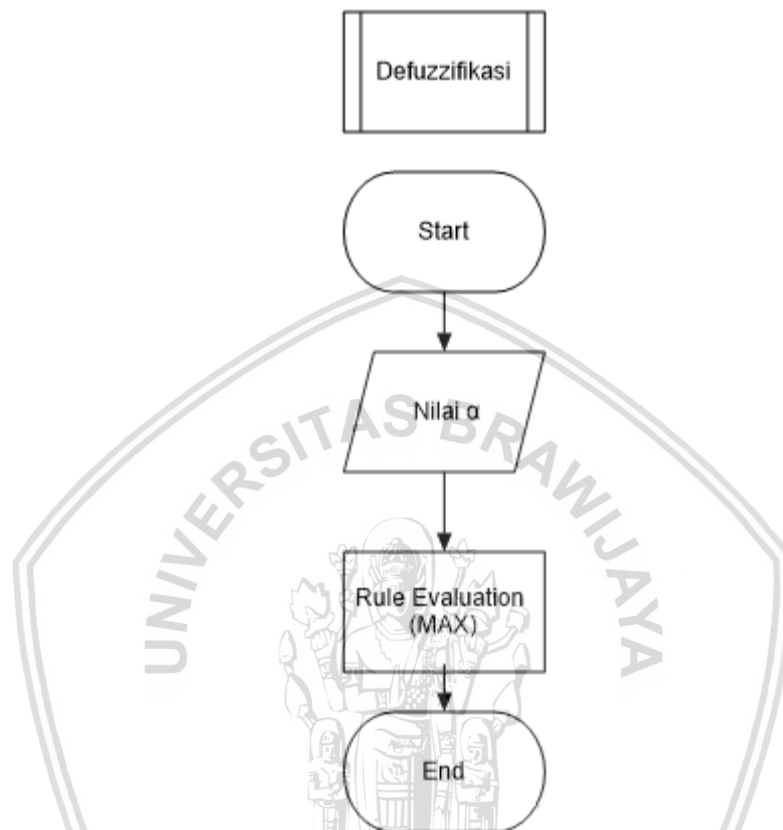
$$\begin{aligned}
 \alpha\text{-predikat}_1 &= \min (\mu_{\text{suhuSejuk}} \cap \mu_{\text{ApiJauh dengan sensor}} \cap \mu_{\text{asap Sedang}}) \\
 &= \min (0,1 \cap 1 \cap 1) \\
 &= 0,1
 \end{aligned}$$

- **RULE[31]: IF SUHU HANGAT && API JAUH DENGAN SENSOR && ASAP SEDANG THEN **BERAPI ASAP SEDIKIT****

$$\begin{aligned}
 \alpha\text{-predikat}_1 &= \min (\mu_{\text{suhuHangat}} \cap \mu_{\text{ApiJauh dengan sensor}} \cap \mu_{\text{asap Sedang}}) \\
 &= \min (0,9 \cap 1 \cap 1) \\
 &= 0,9
 \end{aligned}$$

5.1.2.15 Defuzzifikasi

Defuzzifikasi merupakan proses pemetaan dari himpunan fuzzy ke himpunan tegas. Setelah mendapat nilai α , berdasarkan metode MIN-MAX setiap variabel akan mengevaluasi pada setiap rule untuk mencari nilai terbesarnya (max). berikut ini adalah flowchart proses defuzzifikasi.



Gambar 5.13 Flowchart Proses Defuzzifikasi

Pada gambar 5.13 dapat dilihat proses defuzzifikasi membutuhkan nilai input/masukan berupa nilai α yang merupakan hasil dari proses inferensi yang menghasilkan nilai keanggotaan baru. Setelah nilai α didapatkan, maka langkah selanjutnya adalah *rule evaluation*. Proses *rule evaluation* menggunakan metode MAX dimana nilai terbesar akan menjadi *output* dari sistem. Terdapat 4 output dalam sistem ini berupa defuzzifikasi tidak ada kebakaran, berasap, berapi asap sedikit, dan berapi asap banyak. Berikut merupakan cara pengambilan nilai *max* untuk menentukan output dari sistem.

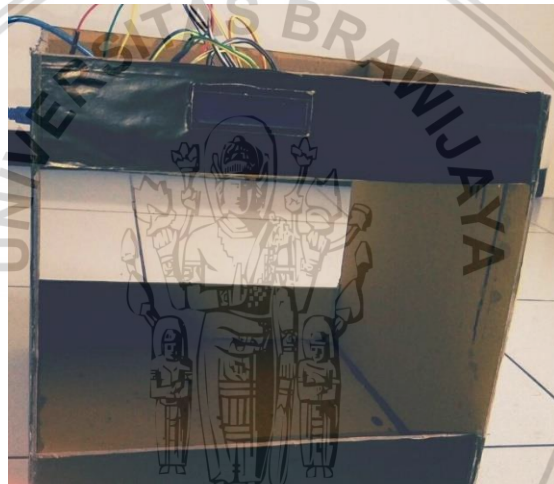
- Tidak ada kebakaran = max (rule0, rule1, rule2, rule3, rule4, rule5, rule6, rule8, rule9, rule10, rule12, rule13, rule13, rule14, rule15, rule16, rule17, rule18, rule20, rule21, rule22, rule24, rule25, rule25, rule27, rule36, rule37, rule38, rule39)
- Berasap = max(rule7, rule11, rule19, rule23, rule31, rule35, rule43, rule47)
- Berapi asap sedikit = max(rule28, rule29, rule30, rule40, rule41, rule 42)
- Berapi asap banyak = max(rule32, rule33, rule34, rule44, rule45, rule46)

5.2 Implementasi Sistem

Pada sub bab implementasi sistem akan menjelaskan tentang implementasi dari sistem berdasarkan dari perancangan sistem pada perangkat keras dan perangkat lunak.

5.2.1 Implementasi Perangkat Keras

Implementasi perangkat keras merupakan hasil dari perancangan dari sistem menjadi sistem yang siap digunakan agar sesuai dengan fungsinya. Sistem ini menggunakan beberapa perangkat keras berupa sensor, mikrokontroller, LCD, dan *buzzer*. Sensor yang digunakan dalam sistem ini yakni sensor suhu DHT11, sensor asap MQ-2, dan sensor *flame* untuk mendeteksi api. Mikrokontroller yang digunakan dalam sistem ini adalah Arduino Mega 2560 untuk melakukan proses logika fuzzy yang dari input yang diberikan, sedangkan LCD dan *buzzer* berfungsi sebagai output dari sistem. Berikut adalah implementasi perangkat keras yang ada pada Gambar 5.14 dan Gambar 5.15



Gambar 5.14 Tampilan prototipe dari depan



Gambar 5.15 Tampilan Prototipe dari dalam

5.2.2 Implementasi Perangkat Lunak

Implementasi perangkat lunak merupakan hasil dari perancangan perangkat lunak yang sudah dijelaskan sebelumnya. Pada bagian ini terdapat beberapa

bagian yaitu program implementasi sensor *flame* sebagai pin interrupt, program inisialisasi untuk menjalankan perintah *low power*, program untuk membaca nilai sensor, pembentukan fuzzifikasi, pembuatan rule hasil kombinasi dari himpunan pada setiap variabel yang sudah ditentukan, melakukan inferensi, dan defuzzifikasi.

5.2.1.1 Implementasi sensor flame sebagai pin interrupt

Pada penelitian ini penulis menggunakan sensor *flame* sebagai pin interrupt yang berfungsi untuk berpindah dari mode *sleep* menjadi mode *wake*. Proses yang pertama adalah mendeklarasikan pin yang akan digunakan sebagai pin interrupt.

```
#define WAKE_PIN (2)
```

Gambar 5.16 Deklarasi Pin Interrupt

Pada gambar 5.16 penulis menggunakan pin 2 digital sebagai interrupt, maka sensor *flame* akan dihubungkan dengan pin 2 digital sebagai interrupt.

5.2.1.2 Implementasi Program Low Power

Pada penelitian ini penulis menggunakan mode sleep mode power down, dimana sistem akan mematikan fungsi yang tidak dibutuhkan yang terdapat pada fungsi *sleepNow()* seperti pada Tabel 5.3.

Tabel 5.3 Kode Program Fungsi Sleepnow()

No	Kode Program
1	void sleepNow()
2	{
3	set_sleep_mode(SLEEP_MODE_PWR_DOWN);
4	sleep_enable();
5	power_adc_disable();
6	power_spi_disable();
7	power_timer0_disable();
8	power_timer1_disable();
9	power_timer2_disable();
10	power_twi_disable();
11	attachInterrupt(0, wakeUpNow, LOW);
12	sleep_mode();
13	sleep_disable();
14	power_all_enable();
15	detachInterrupt(0);
16	}
17	

Pada Tabel 5.3 interrupt akan dijalankan apabila pin interrupt berada pada kondisi LOW dimana ini akan terjadi perpindahan state dari sleepNow berpindah menjadi wakeUpNow.

5.2.1.3 Implementasi Kalibrasi sensor MQ-2

Untuk mendapatkan nilai yang sesuai yakni nilai ppm(part per million) dibutuhkan kalibrasi dengan menggunakan kode program MQGetGasPercentage(MQRead(MQ2PIN)/Ro,GAS_SMOKE). Langkah pertama yang dilakukan adalah dengan mencari Ro. Ro merupakan resistansi dari sensor. Berikut kode program untuk mencari nilai Ro pada Tabel 5.4.

Tabel 5.4 Kode Program Kalibrasi Sensor MQ-2

No	Kode Program
1	Serial.print("Calibration\n");
2	Ro = MQCalibration(MQ2PIN);
3	Serial.print("Calibration Done\n");

Dalam potongan kode program untuk mencari nilai Ro terdapat pemanggilan fungsi yakni MQCalibration. Berikut adalah program fungsi MQCalibration pada Tabel 5.5

Tabel 5.5 Kode Program Fungsi MQCalibration

No	Kode Program
1	float MQCalibration(int mq_pin)
2	{
3	int i;
4	float RS_AIR_val = 0, r0;
5	for (i = 0; i < CALIBARAION_SAMPLE_TIMES; i++) {
6	//take multiple samples
7	RS_AIR_val += MQResistanceCalculation(analogRead(mq_pin));
8	delay(CALIBRATION_SAMPLE_INTERVAL);
9	}
10	RS_AIR_val = RS_AIR_val / CALIBARAION_SAMPLE_TIMES;
11	//calculate the average value
12	r0 = RS_AIR_val / RO_CLEAN_AIR_FACTOR_MQ2;
13	//RS_AIR_val divided by RO_CLEAN_AIR_FACTOR yields the Ro
14	//according to the chart in the datasheet
15	return r0;
16	}

Pada potongan kode program MQCalibration diasumsikan sensor berada pada kondisi udara bersih. Hal itu menggunakan fungsi MQResistanceCalculation untuk kalkulasi resistansi sensor pada udara bersih dan membaginya dengan RO_CLEAN_AIR_FACTOR.

Langkah kedua yakni dengan mendapatkan nilai MQRead. Untuk mendapatkan nilai MQRead diperlukan fungsi MQRead(int mq_pin). MQ_PIN merupakan pin yang dihubungkan pada mikrokontroller untuk membaca nilai ADC pada sensor. Berikut isi program dari fungsi MQRead pada Tabel 5.6. Pada baris 6 sensor MQ-2 akan melakukan sampling pembacaan sebanyak 50 kali sebelum menampilkan hasil pengukuran berupa ppm.

Tabel 5.6 Kode Program Fungsi MQRead

No	Kode Program
1	float MQRead(int mq_pin)
2	{
3	int i;
4	float rs = 0;
5	
6	for (i = 0; i < READ_SAMPLE_TIMES; i++) {
7	rs += MQResistanceCalculation(analogRead(mq_pin));
8	delay(READ_SAMPLE_INTERVAL);
9	}
10	
11	rs = rs / READ_SAMPLE_TIMES;
12	
13	return rs;
14	}

Dalam kode program fungsi MQRead(int mq_pin) terdapat pemanggilan fungsi MQResistanceCalculation untuk kalkulasi *sensor resistance (Rs)*. *Sample time* dan *time interval* dapat dikonfigurasi dengan mengubah definisi macros. Berikut adalah Gambar 5.7 yang merupakan isi kode program fungsi MQResistanceCalculation.

Tabel 5.7 Kode Program Fungsi MQResistanceCalculation

No	Kode Program
1	float MQResistanceCalculation(int raw_adc)
2	{
3	return (((float)RL_VALUE_MQ2 * (1023 - raw_adc) / raw_adc));
4	}

Dalam fungsi MQResistanceCalculation terdapat raw_adc yang menampilkan tegangan. Sedangkan RL_Value merupakan nilai dari *Load resistance*. Langkah selanjutnya mendapatkan kalibrasi nilai ADC pada sensor dengan memanggil fungsi MQGetGasPercentage. Berikut Tabel 5.8 merupakan isi dari fungsi MQGetGasPercentage.

Tabel 5.8 Kode Program Fungsi MQGetGasPercentage

No	Kode Program
1	int MQGetGasPercentage(float rs_ro_ratio, int gas_id)
2	{
3	if (accuracy == 0) {
4	return (pow(10, ((-2.331 * (log10(rs_ro_ratio))) + 3.596)));
5	}
6	
7	else if (accuracy == 1) {
8	return (pow(10, (-0.976 * pow((log10(rs_ro_ratio)), 2) - 2.018 * (log10(rs_ro_ratio)) + 3.617)));
9	}
10	return 0;
11	}
12	

Sensor MQ-2 digunakan untuk membaca intensitas asap didalam ruangan. Berikut Tabel 5.9 yang merupakan rumus untuk kalibrasi nilai ADC dari sensor MQ-2 ke ppm (Part Per Million).

Tabel 5.9 Kode Program Kalibrasi ADC ke PPM

No	Kode Program
1	Serial.print("SMOKE:");
2	Serial.print(MQGetGasPercentage(MQRead(MQ2PIN)/Ro,GAS_SMOKE));
3	Serial.print("ppm");

Untuk mendapatkan pembacaan sensor berupa PPM adalah dengan cara mencari nilai dari MQGetGasPercentage. Nilai dari MQGetGasPercentage didapat dari MQRead(MQ2PIN)/Ro dan GAS_SMOKE. Ro merupakan nilai resistansi dari sensor.

5.2.1.4 Implementasi *Fuzzy Sugeno* pada Sistem

Sebelum melakukan implementasi *Fuzzy Sugeno* diperlukan menginisialisasi *library* pada Arduino IDE, inisialisasi pin yang digunakan dalam mikrokontroller. Dalam mengimplementasi *fuzzy logic* kedalam sistem, langkah pertama yang dilakukan adalah dengan membaca input dari 3 sensor yaitu sensor DHT11, sensor MQ-2, dan sensor *flame*. Pembacaan sensor yang digunakan berupa *output* analog untuk menampilkan data yang akurat agar berfungsi pada pembuatan fuzzifikasi. Berikut potongan kode program fuzzifikasi dari variabel suhu.

Tabel 5.10 Kode Program Fuzzifikasi Suhu

No	Kode Program
1	t = dht.readTemperature();
2	//suhu dingin
3	if (t <= 10.00)
4	{

5	suhu [0] = 1.00;
6	}
7	else if (t > 10.00 && t < 20.00)
8	{
9	suhu [0] = (20.00 - t) / (20.00 - 10.00);
10	}
11	else
12	{
13	suhu [0] = 0.00;
14	}

Pada Tabel 5.10 ditunjukkan proses pembacaan dari suhu menggunakan sensor DHT11 melalui `t= dht.readTemperature();`. Pada Tabel 5.10 merupakan fuzzifikasi terhadap suhu dingin, untuk suhu sejuk, hangat dan panas menggunakan kode yang sama namun mengubah variabel array dengan nilai yang sesuai dengan membership function. Setelah sensor berhasil melakukan sensing maka nilai pembacaan akan dibagi kedalam variabel suhu yaitu suhu dingin,sejuk , hangat, dan panas. Nilai suhu akan disimpan dalam array suhu[4] yakni suhu dingin dalam suhu[0], suhu sejuk dalam suhu[1], suhu hangat dalam suhu[2], dan suhu panas dalam [3]. Langkah selanjutnya adalah membuat fuzzifikasi asap dimana kode program terdapat pada Tabel 5.11

Tabel 5.11 Kode Program Fuzzifikasi Asap

No	Kode Program
1	AsapV1 = analogRead (Asap);
2	AsapV = (MQGetGasPercentage (MQRead (MQ2PIN) / Ro,
3	GAS_SMOKE));
4	//tidak ada asap dan renggang
5	if (AsapV <= 25.00) {
6	asap [0] = 1.00;
7	}
8	else if (AsapV > 25.00 && AsapV <= 50.00) {
9	asap [0] = (50.00 - AsapV) / (50.00 - 25.00);
10	}
11	else {
12	asap [0] = 0.00;
13	}

Pada Tabel 5.11 dijelaskan pembacaan sensor dengan menggunakan analog read ke ppm (part per million). Setelah sensor berhasil melakukan sensing maka hasil pembacaan akan diubah kedalam himpunan dari variabel asap yakni asap renggang, asap sedang, dan asap pekat. Nilai pembacaan asap akan disimpan dalam array asap[3], dimana tidak ada asap akan disimpan dalam asap[0], asap sedang akan disimpan dalam asap[1], dan asap pekat akan disimpan dalam asap[2]. Langkah selanjutnya adalah melakukan fuzzifikasi api yang akan dijelaskan pada Tabel 5.12

Tabel 5.12 Kode Program Fuzzifikasi Api

No	Kode Program
1	ApiV = analogRead(Api);
2	//sensor dekat dengan api
3	if (ApiV <= 60.00) {
4	api [0] = 1.00;
5	}
6	else if (ApiV > 60.00 && ApiV <= 100.00) {
7	api [0] = (100.00 - ApiV) / (100.00 - 60.00);
8	}
9	else {
10	api [0] = 0.00;
11	}

Pada Tabel 5.12 ditunjukkan pembacaan sensor flame dengan menggunakan `ApiV = analogRead(Api);`. Pembacaan sensor akan ditampilkan dalam bentuk analog yang ditampilkan dalam serial monitor. Setelah sensor flame berhasil melakukan sensing maka nilai akan disimpan dalam array api [4]. Dimana api dekat dengan sensor disimpan dalam array api[0], api agak dekat dengan sensor disimpan dalam api[1], api jauh dengan sensor disimpan dalam api[2], tidak ada api disimpan dalam api[3].

Setelah fuzzifikasi dari tiap sensor selesai dibuat, maka langkah selanjutnya adalah dengan membuat rule fuzzy dan inferensi. Proses inferensi dilakukan dengan cara membandingkan nilai pembacaan dari setiap variabel yakni variabel suhu, api, dan asap dengan menggunakan perhitungan min.

Berikut adalah potongan program dalam pembuatan rule fuzzy dan inferensi. Pada Tabel 5.13 terdapat potongan dari total 48 rule yang dibuat berdasarkan masing-masing himpunan pada setiap variabel. Pada Tabel 5.13 aturan yang dibuat dengan menggunakan fungsi implikasi. Pada proses fungsi implikasi yang digunakan adalah min. Dalam menampilkan hasil output terdapat kemungkinan hasil yang sama, maka untuk mengambil salah satu aturan digunakan fungsi max dimana nilai terbesar yang akan diambil sebagai output.

Tabel 5.13 Kode Program Pembuatan Rule

No	Kode Program
1	rule0 = min (suhu[0], min (asap[0], api[0]));
2	rule1 = min (suhu[0], min (asap[0], api[1]));
3	rule2 = min (suhu[0], min (asap[0], api[2]));
4	rule3 = min (suhu[0], min (asap[0], api[3]));
5	rule4 = min (suhu[0], min (asap[1], api[0]));
6	rule5 = min (suhu[0], min (asap[1], api[1]));
7	rule6 = min (suhu[0], min (asap[1], api[2]));
8	rule7 = min (suhu[0], min (asap[1], api[3]));
9	rule8 = min (suhu[0], min (asap[2], api[0]));
10	rule9 = min (suhu[0], min (asap[2], api[1]));
11	rule10 = min (suhu[0], min (asap[2], api[2]));
12	rule11 = min (suhu[0], min (asap[2], api[3]));
13	rule12 = min (suhu[1], min (asap[0], api[0]));
14	rule13 = min (suhu[1], min (asap[0], api[1]));
15	rule14 = min (suhu[1], min (asap[0], api[2]));
16	rule15 = min (suhu[1], min (asap[0], api[3]));
17	rule16 = min (suhu[1], min (asap[1], api[0]));
18	rule17 = min (suhu[1], min (asap[1], api[1]));

19	rule18 = min (suhu[1], min (asap[1], api[2]));
20	rule19 = min (suhu[1], min (asap[1], api[3]));
21	rule20 = min (suhu[1], min (asap[2], api[0]));
22	rule21 = min (suhu[1], min (asap[2], api[1]));
23	rule22 = min (suhu[1], min (asap[2], api[2]));
24	rule23 = min (suhu[1], min (asap[2], api[3]));
25	rule24 = min (suhu[2], min (asap[0], api[0]));
26	rule25 = min (suhu[2], min (asap[0], api[1]));
27	rule26 = min (suhu[2], min (asap[0], api[2]));
28	rule27 = min (suhu[2], min (asap[0], api[3]));
29	rule28 = min (suhu[2], min (asap[1], api[0]));
30	rule29 = min (suhu[2], min (asap[1], api[1]));
31	rule30 = min (suhu[2], min (asap[1], api[2]));
32	rule31 = min (suhu[2], min (asap[1], api[3]));
33	rule32 = min (suhu[2], min (asap[2], api[0]));
34	rule33 = min (suhu[2], min (asap[2], api[1]));
35	rule34 = min (suhu[2], min (asap[2], api[2]));
36	rule35 = min (suhu[2], min (asap[2], api[3]));
37	rule36 = min (suhu[3], min (asap[0], api[0]));
38	rule37 = min (suhu[3], min (asap[0], api[1]));
39	rule38 = min (suhu[3], min (asap[0], api[2]));
40	rule39 = min (suhu[3], min (asap[0], api[3]));
41	rule40 = min (suhu[3], min (asap[1], api[0]));
42	rule41 = min (suhu[3], min (asap[1], api[1]));
43	rule42 = min (suhu[3], min (asap[1], api[2]));
44	rule43 = min (suhu[3], min (asap[1], api[3]));
45	rule44 = min (suhu[3], min (asap[2], api[0]));
46	rule45 = min (suhu[3], min (asap[2], api[1]));
47	rule46 = min (suhu[3], min (asap[2], api[2]));
48	rule47 = min (suhu[3], min (asap[2], api[3]));

Tabel 5.14 Kode Program Defuzzifikasi

No	Kode Program
1	//=====defuzzifikasi=====
2	s1 = max (rule0, max(rule1, rule2));
3	s2 = max(rule3, max (rule4, rule5));
4	s3 = max(rule6, rule8);
5	s4 = max(rule9, rule10);
6	s5 = max(rule12, max(rule13, rule14));
7	s6 = max(rule15, rule16);
8	s7 = max(rule17, rule18);
9	s8 = max(rule20, max(rule21, rule22));
10	s9 = max(rule24, max(rule25, rule26));
11	s10 = max(rule27, rule36);
12	s11 = max(rule37, max(rule38, rule39));
13	normal = max(s1, max(s2, s3));
14	normal1 = max(s4, max(s5, s6));
15	normal2 = max(s7, max(s8, s9));
16	normal3 = max(s10, s11);
17	fixnormal = max (normal, max(normal1, max(normal2, normal3)));
18	w1 = max(rule7, rule11);
19	w2 = max(rule19, max(rule23, rule31));
20	w3 = max(rule35, max(rule43, rule47));
21	waspada = max (w1, max (w2, w3));
22	kebakaran_asap_banyak = max(rule32, max(rule33, rule34));
23	kebakaran_asap_banyak1 = max(rule44, max(rule45, rule46));
24	fixkebakaran_asap_banyak = max(kebakaran_asap_banyak ,
25	kebakaran_asap_banyak1);
26	kebakaran_asap_sedikit = max(rule28, max(rule29, max(rule30,
27	max(rule40, max(rule41, rule42))));
28	defu = max(fixnormal, max(waspada, max(fixkebakaran_asap_banyak,
29	kebakaran_asap_sedikit)));

Setelah proses defuzzifikasi selesai dilakukan, maka hasil dari defuzzifikasi akan masuk ke dalam proses if-else yang berfungsi mengeluarkan output yang sesuai dengan defuzzifikasi ke LCD dan buzzer. Berikut kode program untuk mengeluarkan output ke LCD dan buzzer pada Tabel 5.15 berikut.

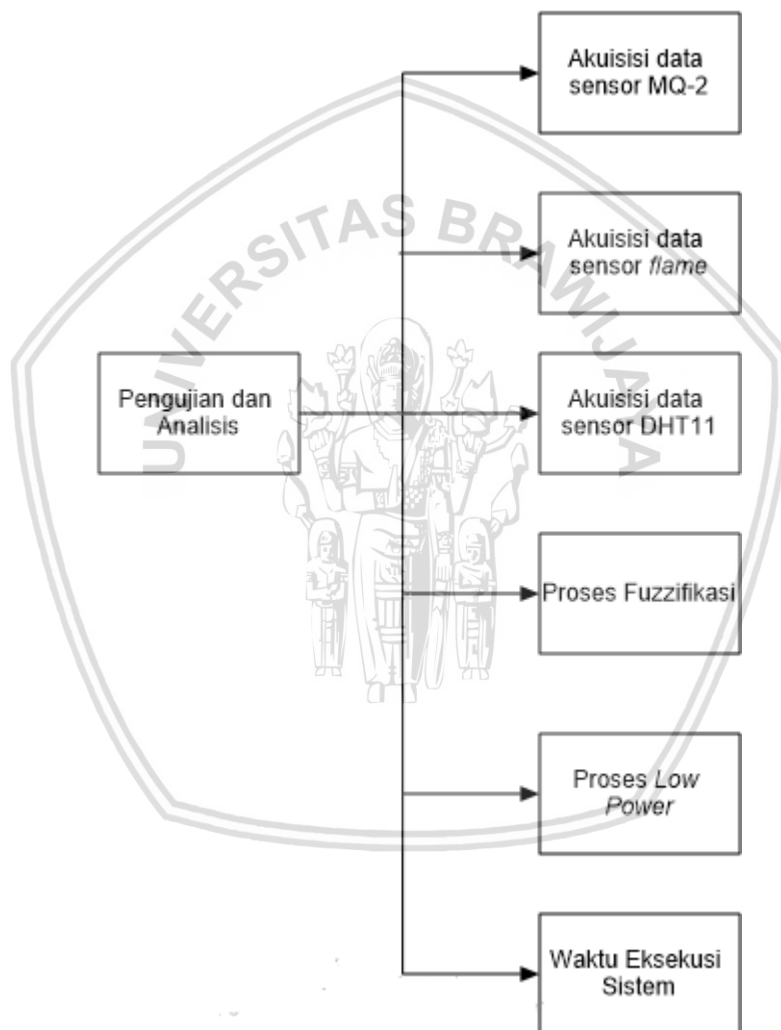
Tabel 5.15 Kode Program Output LCD dan Buzzer

No	Kode Program
1	if (defu == fixnormal) {
2	Serial.println ("OUTPUT: TIDAK ADA KEBAKARAN");
3	lcd.clear();
4	lcd.setCursor(2, 0); //Setting cursor pada LCD
5	lcd.print("CONDITION :"); //Menampilkan "Kondisi" pada LCD
6	lcd.setCursor(0, 2);
7	lcd.print("TIDAK KEBAKARAN");
8	analogWrite(buzz, 0);



BAB 6 PENGUJIAN DAN ANALISIS

Bab pengujian dan analisis membahas tentang proses pengujian pada penelitian tentang “ Implementasi *Low Power* Pada Embedded System Untuk Mendeteksi Kondisi Kebakaran Dalam Ruangan”. Dalam sistem ini terdapat pengujian akuisisi data dari sensor MQ-2 untuk mendapatkan pembacaan data intensitas asap, akuisisi data sensor DHT11 untuk mendapatkan pembacaan data suhu, sensor flame untuk mendapatkan pembacaan data api, pengujian fuzzifikasi, pengujian *low power*, dan pengujian eksekusi waktu. Pengujian dalam sistem ini ditunjukkan pada Gambar 6.1



Gambar 6.1 Diagram Blok Pengujian dan Analisis

6.1 Pengujian Akuisisi Data Sensor MQ-2

6.1.1 Tujuan

Tujuan dilakukannya pengujian akuisisi data pembacaan intensitas asap dari sensor MQ-2 adalah untuk mengetahui kemampuan pembacaan sensor MQ-2 dalam membaca intensitas asap dalam satuan ppm (part per million) yang merupakan hasil kalibrasi dari pembacaan ADC pada serial monitor Arduino IDE. Pengujian dilakukan dengan membuat asap dari pembakaran kertas.

6.1.2 Prosedur

Prosedur yang harus dilakukan dalam pengujian sensor MQ-2 antara lain :

1. Menghubungkan MQ-2 dengan Arduino Mega 2560 menggunakan kabel *jumper* sehingga sensor akan terhubung dengan mikrokontroler sesuai dengan pin yang sudah ditentukan.
2. Buka Arduino IDE kemudian mengetik kode program untuk membaca sensor dan mengkalibrasi *output* yang awalnya berupa nilai resistansi analog menjadi nilai ppm (*part per million*).
3. Compile dan *upload* kode program yang telah dibuat.
4. Mengamati hasil pembacaan sensor melalui serial monitor yang telah disediakan Arduino IDE dan tunggu sampai proses kalibrasi selesai dan hasil pembacaan mulai stabil.
5. Memberikan *input* pada sensor berupa asap yang dihasilkan dari pembakaran kertas.
6. Amati hasil pembacaan dari sensor ketika sensor sudah mendapatkan data dari asap.
7. Kesimpulan

6.1.3 Hasil dan Analisis

Berikut adalah Tabel 6.1 yang merupakan hasil dari pembacaan sensor MQ-2 setelah melalui beberapa percobaan sehingga didapatkan data dari asap, hasil pembacaan di bagi menjadi tiga, yakni ketika tidak ada asap disekitar sensor, asap dengan variasi kepekatan yang berbeda-beda, ketika asap mulai terurai.

Tabel 6.1 Hasil Pengujian Sensor MQ-2 melalui Arduino IDE

No	Kondisi	Nilai Pembacaan Sensor MQ-2 (ppm)
1	Tidak Ada Asap	0
2		0
3		0

Percobaan	Kondisi	Nilai Pembacaan Sensor MQ-2 (ppm)
4	Asap dengan variasi kepekatan yang berbeda	8
5		12
6		32
7		99
8		195
9		262
10		398
11		622
12		807
13	Asap Mulai Terurai	228
14		111
15		105
16		73
17		23
18		19
19		4
20		0

Pada tabel 6.1 menunjukkan bahwa pengujian akuisisi data sensor MQ-2 telah berhasil dilakukan. Pada saat sensor tidak mendeteksi adanya asap di sekitar sensor, hasil nilai pembacaan pada sensor adalah 0 ppm. Ketika asap dari kertas yang telah dibakar didekatkan dengan sensor, maka nilai pembacaan dari sensor dalam bentuk ppm mulai naik, yaitu dari 8, 12, 32, 99, 195, dan seterusnya sampai pada nilai 807 ppm dapat dilihat dari kondisi asap dengan variasi kepekatan yang berbeda mulai dari sumber asap dari jarak jauh sampai pada jarak terdekat dengan sensor. Ketika asap dari kertas dijauhkan mulai dari terdekat sampai pada terjauh dari sensor nilai pembacaan dari sensor mulai mengalami penurunan, yaitu dari 807 menjadi 228, 111, 105 dan seterusnya hingga 0 ppm yang berarti tidak ada asap. Dari hasil pengujian diatas dapat diambil kesimpulan bahwa sensor berhasil dalam mendeteksi asap dengan baik dan dapat digunakan untuk akuisisi data sebagai input.

6.2 Pengujian Akuisisi Data Sensor DHT11

6.2.1 Tujuan

Tujuan melakukan pengujian pada sensor DHT11 adalah untuk mengetahui akurasi pembacaan data dari sensor dengan cara membandingkan nilai yang dihasilkan oleh sensor dengan suhu yang terdeteksi oleh termometer ruangan.

6.2.2 Prosedur

1. Menghubungkan sensor DHT11 dengan Arduino Mega 2560 menggunakan kabel *jumper* sehingga sensor akan terhubung dengan mikrokontroler sesuai dengan pin yang ditentukan.
2. Buka Arduino IDE lalu mengetikkan kode program untuk pembacaan sensor dengan memasukkan library DHT11.h untuk membaca dalam satuan celcius.
3. Compile dan upload kode program yang telah dibuat.
4. Membaca suhu ruangan dengan dimensi 3x3 meter pada jam 11:00.
5. Mengamati hasil nilai suhu dari thermometer ruangan dan hasil pembacaan sensor di serial monitor pada Arduino IDE. Kemudian mengambil 10 sampel pembacaan dari thermometer dan sensor DHT11.
6. Menentukan persentase kesalahan dengan menggunakan rumus persentase kesalahan(%) yang terdapat pada rumus (2.1).
7. Menentukan rata-rata dari kesalahan (%) dengan menggunakan rumus rata-rata yang terdapat pada rumus (2.2) :
8. Kesimpulan.

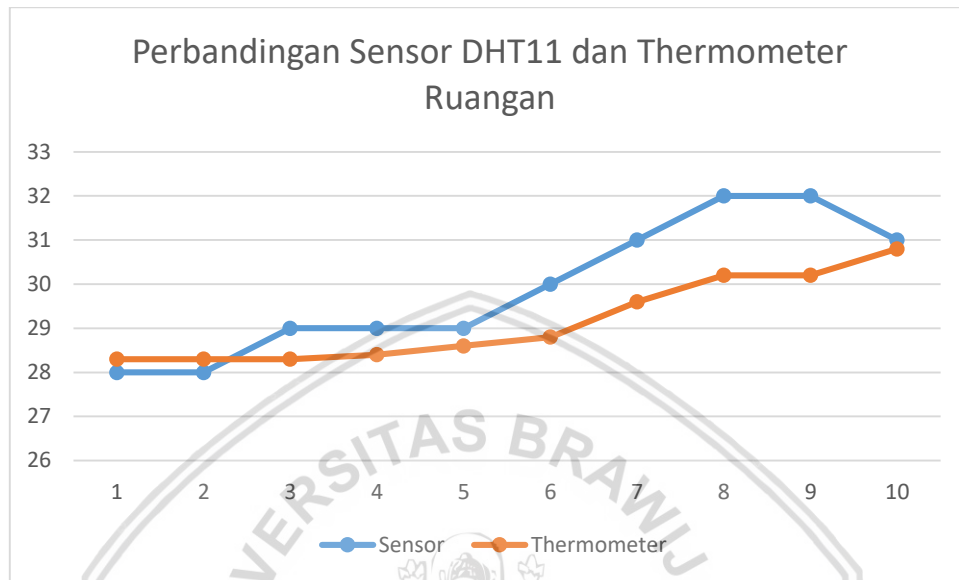
6.2.3 Hasil dan Analisis

Berikut adalah tabel 6.2 yang merupakan hasil pengujian dari sensor DHT11 yang membandingkan sensor dengan pembacaan thermometer ruangan.

Tabel 6.2 Analisis sistem errot pada sensor DHT11

Percobaan	Sensor(C)	Thermometer(C)	Kesalahan
1	28	28,3	1,1%
2	28	28,3	1,1%
3	29	28,3	2,5%
4	29	28,4	2,1%
5	29	28,6	1,4%
6	30	28,8	4,2%
7	31	29,6	4,7%
8	32	30,2	6,0%
9	32	30,2	6,0%
10	31	30,8	0,6%
Rata-rata			3,0%

Berdasarkan data yang ditampilkan pada Tabel 6.2, diketahui hasil pembacaan antara sensor DHT11 dengan thermometer ruangan tidak jauh berbeda. Hal ini dapat dilihat dari nilai rata-rata sistem error sebesar 3,0%. Berikut adalah gambar grafik perbandingan antara pembacaan sensor DHT11 dan thermometer ruangan pada Gambar 6.2



Gambar 6.2 Grafik perbandingan Sensor DHT11 dan Thermometer Ruangan

Pada grafik yang ditunjukkan oleh Gambar 6.2 dapat dilihat bahwa pola pembacaan antara sensor DHT11 dengan Thermometer ruangan tidak stabil namun masih dalam kondisi selisih yang tidak terlalu besar. Pada percobaan pertama dan kedua sensor dan thermometer memiliki selisih yang sedikit, namun ketika pada pembacaan ke 8 dan ke 9 pembacaan sensor DHT11 lebih besar dari thermometer ruangan.

6.3 Pengujian Akuisisi Data Sensor *Flame*

6.3.1 Tujuan

Tujuan dilakukannya pengujian pada sensor flame adalah untuk mengetahui nilai pembacaan ADC yang dihasilkan oleh sensor ketika mendeteksi adanya api.

6.3.2 Prosedur

Pengujian dilakukan berdasarkan jarak antara api dengan sensor. Pada pengujian ini penulis membagi pembacaan jarak sampai 110cm yang dibagi menjadi 11 yaitu jarak 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110 dan tidak ada api. Berikut adalah prosedur yang dilakukan dalam melakukan pengujian sensor flame.

1. Menghubungkan sensor *flame* dengan Arduino Mega 2560 menggunakan kabel *jumper* sehingga sensor dapat terhubung dengan mikrokontroler sesuai dengan pin yang sudah ditentukan.

2. Buka Arduino IDE lalu mengetikkan kode program untuk pembacaan nilai analog dari sensor *flame*.
3. Compile dan Upload kode program yang telah dibuat
4. Amati hasil *output* dari sensor ketika membaca adanya api pada jarak 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110 cm.
5. Kesimpulan.

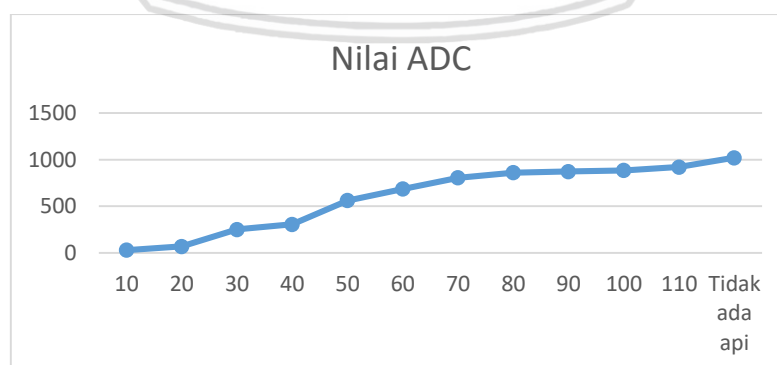
6.3.3 Hasil dan Analisis

Berikut adalah Tabel 6.3 yang merupakan hasil dari pembacaan sensor *flame* :

Tabel 6.3 Nilai Pembacaan Analog pada Sensor *Flame*

Percobaan(cm)	Nilai Pembacaan Serial
10	28
20	68
30	250
40	305
50	561
60	686
70	805
80	861
90	873
100	885
110	921
Tidak ada api	1021

Berdasarkan pada Tabel 6.3 pembacaan nilai analog pada sensor flame akan menurun apabila api semakin dekat dengan sensor dan pembacaan analog yang naik ketika api jauh dengan sensor dan tidak ada api. Berikut adalah Gambar 6.3 yang menunjukkan grafik pembacaan analog pada sensor flame.



Gambar 6.3 Grafik Pengujian Pembacaan Sensor *Flame*

Pada Gambar 6.3 menunjukkan bahwa ketika sensor flame tidak mendeteksi adanya api ketika nilai analog berada pada 921 sampai 1021 namun nilai analog akan semakin menurun apabila api dekat dengan sensor.

6.4 Pengujian Proses Fuzzy

6.4.1 Tujuan

Tujuan dilakukan pengujian proses fuzzy adalah untuk mengetahui ketepatan sistem dalam menentukan hasil pembacaan dari sensor yang kemudian dilakukan perhitungan fuzzy.

6.4.2 Prosedur

Pengujian dilakukan dengan membandingkan perhitungan secara sistem dengan perhitungan secara manual. Sistem dikatakan berhasil melakukan proses fuzzy apabila output hasil perhitungan sistem dengan output hasil perhitungan secara manual serupa.

1. Menyambungkan sensor MQ-2, sensor DHT11, sensor *flame*, LCD 2x16, *buzzer* dan Arduino Mega 2560 dengan menggunakan kabel jumper dengan pin yang sudah ditentukan sehingga menjadi sebuah sistem yang saling terhubung.
2. Buka Arduino IDE, lalu ketikkan kode program agar sistem dapat berjalan dengan menggunakan logika fuzzy.
3. Compile dan upload source program yang telah dibuat.
4. Buka Serial monitor pada Arduino IDE untuk mengamati hasil pembacaan dan perhitungan fuzzy. Apabila hasil pembacaan/ *input* dari sensor belum stabil, maka tunggu beberapa saat sampai pada nilai stabil dikarenakan sensor membutuhkan kalibrasi.
5. Memberikan *input* kepada masing-masing sensor sehingga sensor dapat membaca dan melakukan perhitungan dengan logika *fuzzy*. Penulis melakukan *input* asap dengan menggunakan asap hasil pembakaran kertas, sedangkan untuk *input* api menggunakan korek gas, dan *input* suhu dari kenaikan suhu yang disebabkan oleh asap dan api yang ada didalam prototipe.
6. Amati hasil perhitungan melalui serial monitor di Arduino IDE. Catat sampel sebanyak 10 sampel yang merepresentasikan hasil output yaitu tidak ada kebakaran, berasap, berapi asap sedikit, dan berapi asap banyak.
7. Melakukan perbandingan antara keluaran output yang berupa tampilan kondisi pada LCD dan bunyi pada buzzer dengan hasil perhitungan dari sistem.
8. Kesimpulan.

6.4.3 Hasil dan Analisis

Berikut adalah Tabel 6.4 yang merupakan hasil pengujian perhitungan fuzzy menurut sistem yang dapat dilihat pada serial monitor di Arduino IDE.

Tabel 6.4 Hasil Pengujian Fuzzy

No	Suhu	Asap	Api	Output Sistem	Output
1	33	22	1012	Tidak Ada Kebakaran	LCD = Tidak Kebakaran Buzzer = Off
2	30	33	102	Tidak Ada Kebakaran	LCD = Tidak Kebakaran Buzzer = Off
3	28	125	380	Berapi Asap Banyak	LCD = BAS Buzzer = ON
4	28	88	1011	Berasap	LCD = BERASAP Buzzer = On
5	29	35	1012	Tidak Ada Kebakaran	LCD = Tidak Kebakaran Buzzer = Off
6	27	53	69	Berapi Asap Sedikit	LCD = BAS Buzzer = ON
7	28	55	1013	Berasap	LCD = BERASAP Buzzer = On
8	30	116	254	Berapi Asap Banyak	LCD = BAB Buzzer = ON
9	33	43	426	Berapi Asap Sedikit	LCD = BAS Buzzer = ON
10	36	15	1012	Tidak Ada Kebakaran	LCD = Tidak Kebakaran Buzzer = Off

Pengujian yang dilakukan pada Tabel 6.4 dengan mengambil 10 sampel dengan input dan output yang berbeda yaitu tidak ada kebakaran, berasap, berapi asap sedikit, dan berapi asap banyak. Pengujian dilakukan dengan menganalisis hasil output yang diberikan sistem dengan output yang ditampilkan oleh LCD dan Buzzer. Dapat dilihat pada Tabel 6.4 dari 10 sampel yang di catat, hasil pengujian secara sistem dan output dari buzzer dan lcd memiliki hasil yang sama.

Gambar 6.4 menunjukkan hasil output dari sistem adalah tidak kebakaran dengan nilai himpunan tegas. Hasil output dari sistem memiliki hasil yang sama dengan hasil tampilan pada LCD dan Buzzer. Hal ini menunjukkan bahwa sistem sudah berjalan dengan benar dan menghasilkan output yang sesuai dengan perancangan sistem.

6.5 Pengujian Low Power

Untuk mengetahui apakah proses *low power* berjalan sesuai dengan yang diinginkan maka perlu mengetahui konsumsi arus dari purwarupa sistem dan membandingkan pemakaian daya menggunakan *low power* dengan tidak menggunakan *low power*. Pengukuran dilakukan ketika dalam keadaan sleep dengan menggunakan multimeter kemudian dibandingkan dengan konsumsi arus ketika sistem dalam keadaan wake.

6.5.1 Tujuan Pengujian

Untuk menghasilkan sebuah sistem yang memiliki fitur hemat daya atau *low power* maka sistem dikondisikan pada kondisi sleep untuk mengurangi konsumsi

arus. Agar dapat diketahui hasil dari pengukuran konsumsi arus maka dilakukan pengujian pembacaan pada saat keadaan sleep dan wake dan membandingkannya.

6.5.2 Prosedur

Dalam pengujian *low power* berikut prosedur yang dilakukan :

1. Catu daya pada pin GND dari purwarupa sistem dilepas.
2. Multimeter diatur untuk membaca arus
3. Multimeter dipasang antara GND pada purwarupa sistem dan sisi negative Baterai.
4. Konsumsi arus dibaca pada keadaan *wake*, membaca sensor, proses fuzzifikasi, dan keluaran pada output.
5. Konsumsi arus dibaca pada kondisi *sleep*.
6. Hasil pembacaan dicatat dan direkap ke dalam tabel.
7. Menghitung pemakaian daya selama 35 detik. Untuk menghitung pemakaian daya selama 35 detik terdapat pada rumus (2.3).
8. Hasil dari tabel dirata-rata dan kemudian dicari persentase penurunan dari konsumsi arus.
9. Menghitung penurunan daya pada keadaan *wake* dan *sleep* yang terdapat pada rumus (2.4).
10. Kesimpulan.

6.5.3 Hasil dan Analisis

Hasil dari pengukuran arus menunjukkan ketika sistem dalam keadaan membaca sensor dan menampilkan data, arus yang digunakan sebesar 0,245A atau 245mA dan ketika sistem berpindah ke kondisi sleep penggunaan arus menurun mencapai 0,205A atau 205mA begitupun juga ketika sistem tidak menggunakan *low power* arus yang digunakan sebesar 245mA. Dari hasil pengujian dapat diketahui bahwa konsumsi arus listrik mengalami penurunan sebanyak 0,4A atau 40mA jika dibandingkan dengan sistem ketika tidak menggunakan *low power* dimana sistem hanya menjalankan proses perhitungan fuzzy dan juga ketika membaca sensor dan menampilkan data.

Contoh Perhitungan : $Pemakaian = V \times I \times \frac{n}{3600}$

Pemakaian daya ketika mendeteksi keadaan bahaya (wake)

$$= 6 \times 0,245 \times \frac{35}{3600} = 0,014291667$$

Pemakaian daya ketika tidak mendeteksi keadaan bahaya (sleep)

$$= 6 \times 0,205 \times \frac{35}{3600} = 0,01195833$$

Tabel 6.5 Hasil Pengujian Konsumsi Arus Listrik

No	Arus Kondisi Wake (A)	Pemakaian Daya Wake(A)	Arus Kondisi Sleep (A)	Pemakaian Daya Sleep(A)	Tanpa <i>Low Power</i> (A)
1	0,245	0,014291667	0,205	0,01195833	0,245
2	0,245	0,014291667	0,205	0,01195833	0,245
3	0,245	0,014291667	0,205	0,01195833	0,245
4	0,245	0,014291667	0,205	0,01195833	0,245
5	0,245	0,014291667	0,205	0,01195833	0,245
6	0,245	0,014291667	0,205	0,01195833	0,245
7	0,245	0,014291667	0,205	0,01195833	0,245
8	0,245	0,014291667	0,205	0,01195833	0,245
9	0,245	0,014291667	0,205	0,01195833	0,245
10	0,245	0,014291667	0,205	0,01195833	0,245
Rata-rata	0,245	0,014291667	0,205	0,01195833	0,245

6.6 Pengujian Waktu Eksekusi Sistem

6.6.1 Tujuan

Pengujian waktu eksekusi sistem bertujuan untuk mengetahui kecepatan sistem dalam menjalankan pembacaan sensor dan proses perhitungan fuzzy dan waktu yang dibutuhkan untuk wake.

6.6.2 Prosedur

Pengujian dilakukan sesuai dengan prosedur sebagai berikut :

1. Menambahkan kode program untuk mencatat waktu setiap proses kalibrasi, waktu untuk berpindah ke dalam kondisi sleep, waktu untuk wake.
2. Penguji menjalankan sistem.
3. Hasil yang tampil dicatat dan direkap dalam tabel
4. Kemudian Mencari hasil rata-rata dari 10 data yang dicatat.
5. Kesimpulan

6.6.3 Hasil Pengujian

Tabel 6.6 Hasil Pengujian Waktu

No	Calibrating(ms)	Sleep(ms)	Wake(ms)	Wake to Sensing (ms)	Jumlah (ms)
1	25122	602	43	7470	33237
2	25122	604	46	7030	32802
3	25122	602	39	7600	33363
4	25122	603	40	7680	33445
5	25122	602	38	7390	33152
6	25122	604	43	7570	33339
7	25122	602	42	7340	33106
8	25122	603	44	7640	33409
9	25122	604	47	7520	33293
10	25122	602	42	7930	33696
Rata-rata	25122	602,8	42,4	7517	33284,2

Hasil dari pengujian waktu eksekusi sistem menunjukkan waktu yang diperlukan sistem untuk mengkalibrasi sensor memiliki rata-rata waktu 25,122 detik. Waktu yang dibutuhkan sistem untuk berpindah dari pembacaan data ke status sleep membutuhkan waktu 602,8 milidetik atau 0,602 detik. Sedangkan waktu yang dibutuhkan sistem untuk wake memiliki rata-rata 42,4 atau 0,042 detik. Waktu yang dibutuhkan sistem untuk berpindah dari wake ke pembacaan sistem membutuhkan rata-rata waktu 7517 milidetik. Sehingga total waktu yang dibutuhkan sistem memiliki rata-rata 33284,2 milidetik.

BAB 7 PENUTUP

Bab ini membahas mengenai kesimpulan yang diperoleh dari penelitian yang dilakukan yaitu tentang “Implementasi *Low Power* Pada Embedded System Untuk Mendeteksi Kondisi Kebakaran Dalam Ruangan” dan saran-saran untuk pengembangan topik skripsi atau pengembangan penelitian lebih lanjut.

7.1 Kesimpulan

Kesimpulan yang didapat berdasarkan pengujian dan analisa yang telah dilakukan terhadap tugas akhir ini adalah sebagai berikut.

1. Dalam perancangan prototipe sistem pendeteksi kebakaran dalam ruangan dapat disimpulkan bahwa perancangan telah berhasil dimana Arduino Mega 2560 sebagai mikrokontroler, sedangkan untuk membaca suhu dalam ruangan menggunakan sensor DHT11, kemudian untuk mendeteksi kadar asap dalam ruangan menggunakan sensor MQ-2, dan juga untuk mendeteksi keberadaan api didalam ruangan.
2. Dari hasil perancangan dan implementasi *low power* sistem telah berjalan dengan semestinya sesuai dengan yang di harapkan. Dimana perancangan *low power* menggunakan mode *power down sleep* dengan *external interrupt* untuk berpindah pada kondisi *sleep* menjadi *wake* yang telah di implementasikan kedalam sistem telah berjalan dengan yang diharapkan.
3. Dari hasil pengujian performa yang dilakukan pada sistem yang telah dibuat dapat disimpulkan bahwa kinerja sistem telah berjalan sesuai yang di harapkan. Pada saat tidak ada kenaikan kondisi dari tidak ada kebakaran sistem akan pada *sleep* dan ketika terjadi *interrupt* sistem akan *wake*. Sistem Pada pengujian performa *low power* dapat menurunkan sebanyak 40mA dimana ketika sistem dalam keadaan *wake* menggunakan arus sebesar 245mA dan ketika sistem dalam keadaan *sleep* menggunakan arus sebesar 205mA.

7.2 Saran

Saran yang diberikan penulis untuk pengembangan selanjutnya berdasarkan penelitian yang saat ini dilakukan adalah sebagai berikut:

1. Menggunakan sensor asap dan sensor api yang memiliki sensitivitas yang tinggi karena jika diimplementasikan pada ruangan yang *real* jangkauan dari sensor tidak dapat mencakup seluruh ruangan.
2. Untuk pengembangan selanjutnya dapat menggunakan mikrokontroler yang lebih mendukung untuk mengurangi penggunaan arus.

DAFTAR PUSTAKA

- 3D, C. (2014, April 26). *Arduino Sensor*. Diambil kembali dari Createc 3D: <https://createc3d.com/shop/en/arduino-sensor/660-buy-mq2-gas-sensor-methane-butane-lpg-smoke-sensor-price-offer.html>
- ACEEE. (2016). *The 2016 International Energy Efficiency Scorecard*. Washington: ACEEE.
- Adafruit. (2014). <https://learn.adafruit.com>. Dipetik 11 20, 2017, dari <https://learn.adafruit.com/low-power-wifi-datalogging/power-down-sleep>
- Antony, M. (t.thn.). Pusdiklat Khusus Pelaut Pertamina.
- Arduino. (2008). <https://www.arduino.cc>. Dipetik November 20, 2017, dari <https://www.arduino.cc/reference/en/language/functions/external-interrupts/attachinterrupt/>
- Arduino. (2009). <https://www.arduino.cc>. Dipetik September 5, 2017, dari <https://www.arduino.cc/en/Main/ArduinoBoardMega>
- BasicX. (2005, Juni 25). *BasicX*. Diambil kembali dari BasicX: <http://www.basicx.com/Products/SLCD/2X16LCDoverview.html>
- Co, A. E. (2015, Juli 18). *Ariose Electronics Co*. Diambil kembali dari Ariose Electronics Co: <http://www.ario.com.tw/en/2-2651/manufacture/Piezo-buzzer-fordriver-circuit-built-in-id70455.html>
- Electronics, F. (2012, Desember 25). *Flame Sensor Module*. Diambil kembali dari Future Electronics: http://www.fut-electronics.com/wp-content/plugins/fe_downloads/Uploads/Flame-sensor-arduino.pdf.
- Hanafi, M. H., Setiawan, E., & Hamidi, M. A. (2016). Implementasi Logika Fuzzy Pada Sistem Berbasis Field Programmable Gate Array(FPGA). *Jurnal Teknologi Informasi dan Ilmu Komputer*.
- Homautomation. (2014, April 3). *Homautomation*. Diambil kembali dari Homautomation: www.homautomation.org/2014/04/03/best-ways-to-power-a-arduino-according-to-your-need/
- Ihsannurrahim. (2017). Implementasi Low Power Wearable Device Sebagai Heart Rate Monitor Dengan Metode State Machine . *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*.
- Kusumadewi, S. (2003). Artificial Intelligence (Teknik dan Aplikasinya). *Yogyakarta: Graha Ilmu*.
- Kusumadewi, S., & Hartati, S. (2006). Neuro-Fuzzy: Integrasi Sistem Fuzzy dan Jaringan Syaraf. *Yogyakarta: Graha Ilmu*.

- Meimaharani, R., & Listyorini, T. (2014). Analisis Sistem Inference Fuzzy Sugeno Dalam Menentukan Harga Penjualan Tanah Untuk Pembangunan Minimarket. *Simetris*, 5.
- Miza. (2016). <http://www.caradesainrumah.com>. Dipetik September 5, 2017, dari <http://www.caradesainrumah.com/2016/06/berapa-ukuran-dapur-yang-ideal.html?m=1>
- Miza. (2016, Juni). <http://www.caradesainrumah.com>. Diambil kembali dari <http://www.caradesainrumah.com>: <http://www.caradesainrumah.com>
- MySensors. (2017, Februari 4). *Air Humidity Sensor*. Diambil kembali dari MySensors: <https://www.mysensors.org/build/humidity>
- Nallani, S., & Hency, V. B. (2015). Low Power Cost Effective Automatic Irrigation System. *Indian Journal of Science and Technology*.
- Nebath, E. (2014). Rancang Bangun Alat Pengukur Gas Berbahaya CO Dan CO2 di Lingkungan Industri. *E-Journal Teknik Elektro dan Komputer*.
- Nurromianto, I. (2015). *RANCANGAN PROTOTIPE PENDETEKSIAN DINI LOKASI KEBAKARAN BERBASIS WIRELESS DENGAN FUZZY LOGIC DAN PEMBERITAHUAN VIA SMS*. JEMBER.
- Pacific, S. (2017, Agustus 3). *Saberindo Pacific*. Diambil kembali dari Saberindo Pacific: <http://saberindo.co.id/2017/08/03/teori-segitiga-api/>
- Perda. (1992). *Penanggulangan Bahaya Kebakaran Dalam Wilayah Daerah Khusus Ibukota Jakarta*. Jakarta: DKI Jakarta.
- Prayogi, S., Yamin, M., & Ramadhan, R. (2016). Perancangan Dan Implementasi Prototipe Sistem Pendeteksi Asap Dan Panas Pada Ruangan Tertutup Menggunakan Logika Fuzzy Metode Sugeno. 2.
- Purnomo, R. A. (2017). Implementasi Metode Fuzzy Sugeno Pada Embedded System Untuk Mendeteksi Kondisi Ruangan Dalam Ruangan. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*.
- Transmigrasi, T. K. (2011). *Nilai Ambang Batas Faktor Fisika dan Faktor Kimia di Tempat Kerja*. Jakarta: Kementerian Tenaga Kerja dan Transmigrasi .
- Widyantara, H. (2008). Pendeteksian Dan Pengaman Dini Pada Kebakaran Berbasis Personal Computer (PC) Dengan Fuzzy Logic. 10.
- Yusuf, Y. (2017). *metro.sindonews.com*. Dipetik September 5, 2017, dari <https://metro.sindonews.com/read/1235866/170/si-jago-merah-mengamuk-di-clincing-remaha-20-tahun-tewas>